Josip Stjepandić · Nel Wognum
Wim J.C. Verhagen *Editors*

Concurrent Engineering in the 21st Century

Foundations, Developments and Challenges

Springer

Concurrent Engineering in the 21st Century

Josip Stjepandić · Nel Wognum
Wim J.C. Verhagen
Editors

Concurrent Engineering in the 21st Century

Foundations, Developments and Challenges

Springer

Editors
Josip Stjepandić
3D Product Creation
PROSTEP AG
Darmstadt
Germany

Wim J.C. Verhagen
TU Delft
Delft
The Netherlands

Nel Wognum
Social Sciences Department, Management Studies Group
Wageningen University
Wageningen
The Netherlands

ISBN 978-3-319-13775-9 ISBN 978-3-319-13776-6 (eBook)
DOI 10.1007/978-3-319-13776-6

Library of Congress Control Number: 2014956561

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

The editors of this book wish to dedicate its revenues to foundations that support high-potentials in achieving goals that would not have been possible otherwise. Referring to biographies of great scholars, with their achievements and contributions to humanity and science, we often do not realize that many of them experienced financial problems during their studies, which only could be solved with scholarships and grants. One of such foundations is the Foundation Fra Bonifacije Ivan Pavletić (www.zaklada.biskupija-sisak.hr) from Sisak, Croatia, which is dedicated to financially aiding high-potentials with heavy financial needs.

FOUNDATION
fra Bonifacije Ivan Pavletić

Acknowledgments

The editors, Josip Stjepandić, Nel Wognum, and Wim J.C. Verhagen, sincerely thank the Board of Management of PROSTEP AG for its permission to produce the book by providing time and resources. Special thanks also go to PROSTEP colleagues who have provided support in adapting phrasing and English and improving graphical displays, in particular Ulrike Langer, Jonas Moeljadi, and Hannes Volz. They also thank their families for their support for spending so much time to the book.

Contents

Editors and Contributors

About the Editors

Dr. Josip Stjepandić is the Head of business unit 3D Product Creation at PROSTEP AG, the leading product data integration company worldwide. After receiving his grade as M. Eng. from the University of Zagreb and Ph.D. from the University of Technology of Graz, he worked for two automotive suppliers in the areas of engineering simulation and design methodology. From 1994 to 1996, he was Associated Professor for Applied Informatics in Mechanical Engineering at the University of Applied Sciences Dortmund. Since 1996, he has been in charge of consultancy and solution development at PROSTEP AG in the areas of design methodology, supplier integration, systems engineering, knowledge-based engineering, product data validation and visualization, configuration management, and CAD data exchange for many industries (automotive, aerospace, shipbuilding, machinery). One of his most important contributions is the multiply awarded global supplier portal OpenDESC.com. His current research is focused on systems engineering, modular design, and digital factory.

Dr. Nel Wognum received her Master's degree in Medical Informatics from Leiden University and her Ph.D. in Knowledge-Based Systems from University of Twente in The Netherlands. She has been Assistant Professor in University of Twente, first in knowledge-based systems, later in organization and management studies. Her field of interest has since long been the interaction between various dimensions, disciplines, and functions, as in Concurrent Engineering. She was the President of ISPE from 2004 till 2006. Since 2007 she worked as a researcher in Wageningen University in the Management Studies Group in the field of supply chain management in the food area. Concurrent Engineering principles also apply to this field of study. She has authored and co-authored various publications in this field. Since the end of 2014, Nel Wognum is enjoying her retirement.

Dr. Wim J.C. Verhagen is Assistant Professor at the section of Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology. He obtained his M.Sc. and Ph.D. degrees at the same university. The focus of his early research has been on knowledge-based engineering systems in aircraft lifecycle processes. In recent years, Dr. Verhagen's research has focused on aircraft maintenance research, including the development of Maintenance Information System (MIS) functionality using knowledge-based techniques. Furthermore, maintenance execution, spare parts management and logistics, as well as maintenance scheduling optimization are specific topics of interest. He has authored and co-authored a range of publications in the fields of knowledge-based engineering systems and aircraft maintenance.

Contributors

Pablo Bermell-Garcia EADS Innovation Works, Bristol, UK

Alain Biahmou EDAG Engineering AG, Fulda, Germany

Cees Bil RMIT University, Melbourne, VIC, Australia

Milton Borsato Federal University of Technology, Parana, Brazil

Osiris Canciglieri Junior Polytechnic School, Graduate Program in Production Engineering and Systems (PPGEPS), Pontifical Catholic University of Paraná (PUCPR), Paraná, Brazil

Chun-Hsien Chen School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, Singapore

Kevin W.P. Chen Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan

Nai-Feng Chen School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, Singapore

Richard Curran Faculty of Aerospace Engineering, Air Transport and Operations, Delft University of Technology, Delft, The Netherlands

Catarina Ferreira da Silva LIRIS, UMR5205 CNRS, University of Lyon 1, Villeurbanne, France

Rudolf Dotzauer Continental, Ingolstadt, Germany

Jožef Duhovnik Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia

Nicolas Figay EADS, Suresnes, France

Alain-Jérôme Fougères Université de Technologie de Belfort-Montbeliard, Belfort, France

Arnulf Fröhlich PROSTEP AG, Darmstadt, Germany

Parisa Ghodous LIRIS, UMR5205 CNRS, University of Lyon 1, Villeurbanne, France

Matthias Grau PROSTEP AG, Darmstadt, Germany

Sebastian Handschuh Daimler AG, Böblingen, Germany

Jannicke Baalsrud Hauge Bremer Institut für Produktion und Logistik GmbH (BIBA), University of Bremen, Bremen, Germany

Kazuo Hiekata University of Tokyo, Kashiwa, Chiba, Japan

C.T. Hsiao Department of Economics, Tunghai University, Taichung, Taiwan

Ricardo Jardim-Goncalves New University of Lisbon, Uninova, Lisbon, Portugal

Alfred Katzenbach Katzenbach Executive Consulting, Gaienhofen, Germany; University Stuttgart, Stuttgart, Germany

Li Pheng Khoo School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, Singapore

Martin Kurth RAYCE EURL, Lörrach, Germany

Lutz Lämmer PROSTEP AG, Darmstadt, Germany

Laurent Levy Airbus SAS, Toulouse, France

Harald Liese PROSTEP AG, Darmstadt, Germany

Penny H.Y. Liu Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan

Zoran Lulić Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb, Croatia; Adolo 7 d.o.o., Zagreb, Croatia

Ivan Mahalec Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb, Croatia; Adolo 7 d.o.o., Zagreb, Croatia

Bryan R. Moser Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA, USA; Global Project Design, Boston, MA, USA

Maria Lucia Miyake Okumura Polytechnic School, Graduate Program in Production Engineering and Systems (PPGEPS), Pontifical Catholic University of Paraná (PUCPR), Paraná, Brazil

Egon Ostrosi Université de Technologie de Belfort-Montbeliard, Belfort, France

Jerry J.R. Ou Department of Business Administration, Southern Taiwan University, Tainan, Taiwan

Margherita Peruzzini Indesit Company S.p.A, Fabriano AN, Italy; Department of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, Ancona, Italy

Roberto Riascos PROSTEP AG, Darmstadt, Germany

Georg Rock Trier University of Applied Sciences, Trier, Germany

Stefan Rude BMW AG, Munich, Germany

Goran Šagi Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb, Croatia; Adolo 7 d.o.o., Zagreb, Croatia

Andreas Schreiber BMW AG, Munich, Germany

Michael Sobolewski United States Air Force Research Laboratory, WPAFB, Dayton, OH, USA; Polish-Japanese Institute of Information Technology, Warsaw, Poland

Josip Stjepandić PROSTEP AG, Darmstadt, Germany

Jože Tavčar Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia

Karsten Theis PROSTEP AG, Darmstadt, Germany

Mirko Theiss PROSTEP AG, Darmstadt, Germany

Klaus-Dieter Thoben Bremer Institut für Produktion und Logistik GmbH (BIBA), University of Bremen, Bremen, Germany

Amy J.C. Trappey Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan

Charles V. Trappey Department of Management Science, National Chiao Tung University, Hsinchu, Taiwan

Jacques Trienekens Social Sciences Department, Management Studies Group, Wageningen University, Wageningen, The Netherlands

Dietmar Trippner dreiconsult GbR, Landsberied, Germany

Wim J.C. Verhagen Faculty of Aerospace Engineering, Air Transport and Operations, Delft University of Technology, Delft, The Netherlands

Stefan Wiesner Bremer Institut für Produktion und Logistik GmbH (BIBA), University of Bremen, Bremen, Germany

Patrick Wischnewski Logic4Business GmbH, Saarbrücken, Germany

Nel Wognum Social Sciences Department, Management Studies Group, Wageningen University, Wageningen, The Netherlands

Ralph T. Wood Global Project Design, Boston, MA, USA

Robert Ian Marr Young Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire, UK

Xiaojia Zhao Faculty of Aerospace Engineering, Air Transport and Operations, Delft University of Technology, Delft, The Netherlands

Used Trademarks

3CAD evolution	3CAD Consortium
Actify SpinFire	Actify
Altair Data Manager	Altair
ANSYS Engineering Knowledge Manager	ANSYS
Autodesk Streamline	Autodesk
CATIA	Dassault Systemes
Creo	Parametric Technology
DomusPlanner	Domus Planner
Enovia	Dassault Systemes
Google Sketch-up	Google
Google Sketch-up PRO	Google
iSIGHT	Dassault Systemes
LMS OPTIMUS	Siemens PLM Software
MATLAB/SIMULINK	MathWorks
Metron	Tesy Software
Mobilia	CE soft
NX	Siemens PLM Software
Oracle Autovue	Oracle
Pro/ENGINEER	Parametric Technology
SAP PLM	SAP
SimManager	MSC
SolidEdge	Siemens PLM Software
SolidWorks	SolidWorks
Sweethome3D	eTeks
Teamcenter	Siemens PLM Software
Windchill	Parametric Technology

Chapter 1
Introduction to the Book

Josip Stjepandić, Nel Wognum and Wim J.C. Verhagen

Abstract Concurrent Engineering (CE) was conceived as an important concept in the '80s of the previous century. It has been studied and practiced extensively since then in many forms and under various names. Although the term CE is not in frequent use any more these days, the concept has grown, both conceptually and in importance. Actually, the concept has become a precondition for current ways of working in complex, dynamic, projects, supply chains and networks. In this book, the concept of CE is explored both in research and in practice. Both history and the current situation are treated including the many still existing theoretical and practical challenges. This chapter provides an introduction to the book.

Keywords Concurrent engineering · Integrated product development and design processes · Time-to-Market · Collaborative/engineering processes · CE system · ISPE

1.1 Introduction

Concurrent Engineering (CE) is a comprehensive, systematic approach to the integrated, concurrent design and development of complex products and their related processes, including marketing, manufacturing, logistics, sales, customer

J. Stjepandić (✉)
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: josip.stjepandic@prostep.com

N. Wognum
Social Sciences Department, Management Studies Group, Wageningen University, Hollandseweg 1, 6706KN Wageningen, The Netherlands
e-mail: wognumnel@gmail.com

W.J.C. Verhagen
Air Transport and Operations, Aerospace Engineering, TU Delft, 2600 AC Delft, The Netherlands
e-mail: w.j.c.verhagen@tudelft.nl

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_1

support, and disposal. Its goals are higher productivity and lower costs by shorter product development time and also shorter time-to-market. All stakeholders are forced to consider very early all elements of the product life cycle, from conception to disposal, including cost, quality, and time.

CE is a long-term business strategy too, which promises and provides long-term benefits to business, when properly implemented. It shapes an agile, flexible organization to achieve long-term competitive advantage. The main pre-requisites of CE are seamless communication between stakeholders, their willingness and ability to cooperate, and tremendous support by information and communication technology (ICT).

CE is not a method, nor a tool. It is a concept, a way of thinking, requiring many methods and tools to realize. Although the term, coined in the 1980s, has long been used, it has been replaced by many other terms indicating collaboration and information exchange between various disciplines and functions, and different cultures. As such, CE requires a socio-technical approach in which the social environment is taken into account wherein the product and process development process of CE takes place. There is massive interaction between this social environment and the CE process.

Implementing CE in practice is a long-term process, because it requires many organisational and technical skills that are not easy to acquire. An organisation needs to gradually move from a sequential way of working to a more parallel way of working, requiring much more interaction and information exchange between people from different departments or companies. In particular, the culture for information sharing and collaboration has to be gradually developed. This challenge needs to be tackled in an entire supply chain, including small and medium-sized enterprises (SME). Usually, development towards CE is initiated and controlled by the Original Equipment Manufacturer (OEM). The emergence of ICT has tremendously accelerated this implementation.

In the past decades CE was subject of intensive research and development activities. An active community is the "International Society for Productivity Enhancement, Inc." (www.ispe-org.net), which encompasses the work of many researchers and industry experts around the world. In 1994, ISPE introduced the annual international conferences on Concurrent Engineering, which have become formative meetings for a community of people from many countries all over the world. Researchers and senior experts from this community meet every year to share their experiences and discuss current issues on Concurrent Engineering. Many subjects and applications have been discussed during these meetings, including their developments and challenges.

In the past quarter of a century CE has evolved to a mandatory engineering approach in many industries. Looking back to 25 years of continuous development, the editors have concluded that now is the right time to summarize the achievements and current challenges of CE. That is the main idea behind this book. It presents the gradual evolution of the concept of CE and the many technical and social methods and tools that have been developed, including the many theoretical and practical challenges that still exist. Readers are expected to gain a comprehensive picture of the "common sense" of CE as researched and practiced in different world regions and application fields.

1.2 Origins of the Book

This book began as a result of exchange of emails during the preparation of the nineteenth ISPE international conference on CE in Trier, in September 2012. Triggered by the high number of valuable submissions, the conference chairs of that time were forced to reorganize the initial structure of the conference and the proceedings book as well. A suitable introduction into the excellent content was necessary. Just coinciding with the deadline, a late submission came in, entitled "Current concurrency in research and practice". This submission (now part of Chap. 2) was the ideal introduction to the extensive proceedings book, which as of 2014 counts to the 25 % best downloaded eBooks at Springer Verlag [1]. During the conference the idea of a book was discussed for extracting the most valuable content, while revising and expanding it. The content would be casted in an edited book to be published in celebration of the twentieth anniversary of the initial conference on Concurrent Engineering, held in Pittsburgh, USA, in 1994. Josip Stjepandić and Nel Wognum took the editor's role. Later Wim Verhagen was added into the editor's circle. The contributors were primarily recruited from the ISPE community. To gather higher practical relevance, several industry experts were invited to contribute with their best practices. In sum, the contributor's list has grown up to 59 contributors from almost all world regions with relevant CE practice. The composition of this circle is expected to ensure the aforementioned "common sense" of CE.

During the emergence of this book, two more ISPE conferences were held in Melbourne (Australia) and Beijing (China). Taking advantage of this situation, the content of the book comprises references to the most recent achievements published in the proceedings books of these conferences [2, 3]. For sake of completeness, the editors want to take this opportunity and indicate three early publications of CE, namely by Backhouse and Brookes [4], Prasad [5], and Hartley [6]. These publications especially show the aims and practice of CE in its early days.

1.3 Goals of the Book

This book is an attempt to present the latest developments and best practices of the principles of Concurrent Engineering. The presentation includes not only current CE processes and methods, but also, very importantly, complex real-life applications and experiences. These applications and experiences are aimed to show that CE is an indispensable part of business nowadays. Of course, the term CE covers a variety of approaches that can be classified as CE approaches. Each such approach must be connected to an innovation or product and process development. Each approach must also consist of methods and tools to enable and support extensive collaboration and information exchange between people from different disciplines, functions, departments or companies.

The first goal of the book is to describe the "state-of-the-art", summarizing CE achievements. A second goal of the book is to illustrate the choices that exist in organizing information. These choices encompass selection of methods and tools, technical as well as organisational. The methods and tools show the variety of problems that need to be tackled in practice. They should support trade-offs and finding (near-)optimal solutions. The third goal of this book is to demonstrate that CE has become indispensable, used widely in many industries and that the same basic engineering principles can be applied to new, emerging fields like sustainable mobility. The final goal of the book is to provide sufficient examples and use cases that thoroughly illustrate achievements and practices of CE. In addition, many remaining challenges in research and practice have been listed.

1.4 Audience

The authors intend this book to be useful for several audiences: industry experts, managers, students, researchers, and software developers. The content is intended to serve both as an introduction to development and assessment of novel approaches and techniques of CE and as a compact reference for more experienced experts. In this role practitioners can use the content to improve their core competencies and use it as a reference during their daily work. Graduate and undergraduate students who have already mastered several basic areas of engineering may find it useful instruction material to practices in modern industrial product creation processes. Researchers can find recent achievements and challenges in various fields of CE.

Engineers in various design domains, such as mechanical, electrical, computer science, and environmental and logistics engineering may find this book helpful to understand the fundamental background as captured in modern product and process development. It may help them to understand the multi-disciplinary, multi-dimensional and multi-level nature of CE. It may help them to request information they need from and to supply information needed for product and process development to the relevant stakeholders. It will help stakeholders from various domains to understand how CE works and to participate in CE teams.

Managers need to understand information representing numerous facets of CE for developing a comprehensive strategy and establishing suitable engineering structures and organization. The decisions they make must advance the business competitively by meeting quality, cost and time targets. Management and engineering need to exchange information rapidly and seamlessly so that the processes will be adjusted to support the business strategy and so that management can understand and track product issues and maturity. This book present several methods for organising, transferring, tracking and tracing of information.

Students and researchers in the wide area of engineering need comprehensive information on recent achievements and on directions for future research. The book fulfills this need. For this purpose valuable information can be found in the closing part of this book.

Finally, a further audience may consist of developers of tools and development platforms who usually have a strong software engineering background and are not experienced in applications and process development. In particular for those who define and implement integration scenarios, this book could be a useful reference.

1.5 Structure of the Book

The present edited book is a collection of 28 chapters written upon invitation from the editors by internationally recognized experts from academia and industry. Singular chapters contribute to various aspects of basic concepts, methods, technologies, industrial applications, and current challenges of CE. The volume is organized in four parts according to the main subjects: Foundations, New developments and methods, Applications and Current challenges. The structure of the book is illustrated in Fig. 1.1.

The first part of the book is devoted to theoretical issues. Is begins with the early ideas, economic drivers and socio-technical relationships. This includes demonstration of the technological and organisational evolution in the field of CE. The reader may get insight into the background and foundation of CE.

The second part of the book concerns different aspects of new developments and methods which are included or adopted into CE. These methods include technology-oriented approaches (amongst others, Knowledge-based Engineering, Product Lifecycle Visualization, Reverse Engineering, Digital Mock-Up (DMU)) as well as organisation-oriented approaches (e.g. Collaborative Engineering, Systems Engineering).

The third part addresses the most well-known applications of CE in the industry. It collects achievements and experiences in aviation, automotive, machinery, shipbuilding, consumer good industry, medical equipment industry and environmental engineering.

The interference between parts 3 and 4 is intentional, because the new developments and methods are often related to certain industries and, vice versa, certain industries require particular methods to fulfill their specific needs. In each chapter of these both parts, theoretical, technological and organizational foundations and applications are explained by examples ("use cases") from industrial practice to give a comprehensive picture of the practical importance of CE.

The fourth part presents current challenges that have been identified by the editors and authors of the book.

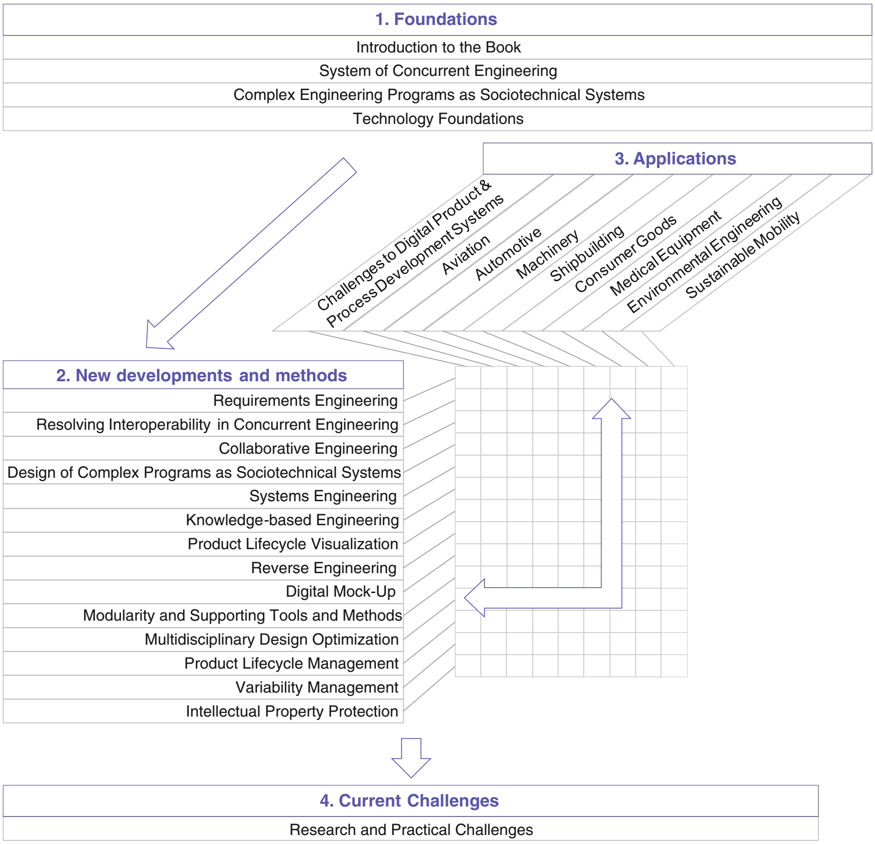


Fig. 1.1 Structure of the book

1.6 Content of the Book

The book has been structured into four consecutive parts, which will be introduced below.

1.6.1 Part 1: Foundations

This Chapter is followed by Chap. 2, in which Nel Wognum and Jacques Trienekens give insight into the gradual evolution of the concept of CE. It shows the growth of the concept from a relatively narrow, but already challenging, concept into an umbrella concept encompassing many socio-technical challenges to tackle. The chapter ends with a systematic approach for framing and describing CE processes, which is applied to an example from the food industry.

In Chap. 3 Bryan Moser and Ralph Wood describe CE processes as complex programs, which require a socio-technical systems approach. They introduce the background of an approach, called *Project Design*, for modeling projects and to simulate likely outcomes, which is further explained in Chap. 8.

In Chap. 4 Michael W. Sobolewski illustrates from the point of view of computer science the technological evolution of approaches to resolving complex multi-level engineering problems concurrently. He focuses on the underlying concepts of CE technology from the point of view of concurrent process expression and its actualization. An architecture of a CE system is derived from the point of view of evolving methodologies of remote method invocation. Within the architecture the integration of concurrent distributed processes, humans, tools and methods to form a trans-disciplinary CE environment for the development of products is presented in the context of cooperating processes and their actualization. Evolving domain-specific languages and service-oriented platforms reflect the complexity of computing problems we are facing in trans-disciplinary CE processes. An architecture of a service-oriented computing environment is described with a service-oriented programing and a coherent operating system for trans-disciplinary large-scale computing. Finally, the SORCER platform is introduced as a solution to the multi-level problems that exist in trans-disciplinary computing systems. A real-life application of the SORCER platform in the defense industry concludes this chapter.

1.6.2 Part 2: New Developments and Methods

In Chap. 5 Stefan Wiesner, Margherita Peruzzini, Jannicke Baalsrud Hauge, and Klaus-Dieter Thoben discuss the topic of Requirements Engineering (RE) as a key to success or failure of every product, service or system development project. An increasingly complicating factor is that from a systems engineering point of view, RE has to define requirements for a rising amount of tangible and intangible components from a growing number of different stakeholders. Additionally, RE has to take into account requirements from every stage of the system life cycle and feed the results back to the development process. Many organizations are still missing effective practices and a documented RE process to tackle the upcoming challenges in systems engineering. The authors give an overview on the RE context and challenges for systems engineering and subsequently describe the state-of-the-art for structuring and processing requirements. They present two case studies illustrate the current situation and methods for resolution in industry. They show how the identified challenges can be met by IT support. The chapter ends with future trends and needs for RE research and challenges for its further integration with CE and life cycle management approaches.

In Chap. 6 Nicolas Figay, Catarina Ferreira da Silva, Parisa Ghodous, and Ricardo Jardim-Gonçalves address the important topic of interoperability. In virtual organisations, like an extended enterprise, with a large degree of sub-contracting and outsourcing, coordination of activities of all partners is necessary, as well as integration of the results of these activities. In such complex emerging networked organizations, it is increasingly challenging to exchange, share and manage internal and external resources like digital information, digital services and computer-enabled processes. The authors characterize expected interoperability for collaborative platform systems and highlight interoperability issues and barriers not yet addressed. They describe an innovative approach for building interoperability based on a Federated Framework of legacy eBusiness standards and address important issues related to semantic preservation along the lifecycle of artifacts and infrastructures required to define and exploit an application. They present two use case studies that apply interoperability strategies.

In Chap. 7 Milton Borsato and Margherita Peruzzini introduce the concept of collaboration and the contexts in which collaboration for CE usually takes place. They pay attention to tools that have been developed to support collaboration, like decision support systems and human-computer interaction. They also describe more recent developments in web-based tools, including the cloud. A basic need for collaboration is explored: the formalization of knowledge as part of information management during a product's lifecycle. During a product lifecycle product models change as well as the need for collaboration support and information management tools. They present two case studies that illustrate the different needs for knowledge formalization, collaboration, and information management.

In Chap. 8 Bryan Moser and Ralph Wood introduce the *Project Design* approach. This approach helps cross-functional teams to explore and converge across scenarios during planning workshops. In this way, project members will become sensitive to the characteristics of the project and create insights into feasibility, value, and a trade space of likely outcomes. Because the people participating in the workshop will actually work together during the CE process many possible barriers for collaboration may have been removed upfront. Two cases illustrate the benefits of the approach for more efficiently and effectively setting up a project.

In Chap. 9 Alain Biahmou discusses product complexity of current cars that have become very complex mechatronic systems that integrate sub-systems created in a synergy between people from different domains such as mechanical engineering, software engineering and electric and electronics (E/E). Product complexity is multidimensional and consists of product, process, organizational, market as well as use complexity. A methodology for mastering complexity is systems engineering, which actually means applying systems thinking to tackle the challenges of creating complex products. The chapter is aimed at providing a deep understanding of systems engineering as well as concepts for its implementation. For this purpose, processes, tools and methods of Systems Engineering are presented. Additionally, a proposal for an introduction process for systems engineering as well as for functional features for practicing systems engineering are presented and discussed.

In Chap. 10 Josip Stjepandić, Wim J.C. Verhagen, Harald Liese, and Pablo Bermell-Garcia discuss the handling of resource knowledge, in particular, company-specific product and process knowledge, as a key to competitiveness. They describe Knowledge-Based Engineering (KBE) as an occurrence of knowledge management. It facilitates new product development by automating repetitive design tasks through acquisition, capture, transform, retention, share, and (re-) use of product and process knowledge. The chapter starts with the definition of knowledge in an engineering context and subsequently addresses the state-of-the-art in KBE research. Three particular areas of research are discussed in detail: knowledge structuring, maintainability of knowledge and KBE applications, and the technological progress and weaknesses of commercial KBE applications like KBE templates. From case study examples, various recent developments in KBE research, development and industrial exploitation are highlighted including scenarios for significant time saving. Challenges for future research and practice have been included.

In Chap. 11 Alfred Katzenbach, Sebastian Handschuh, Rudolf Dotzauer, and Arnulf Fröhlich discuss lifecycle visualization as a rising discipline within product lifecycle management (PLM). Often, when products are developed in 3D using engineering applications, the data is initially stored in the format of the CAD software used. Sharing these data with people who do not have this software or combining them with visualization data from other sources requires a neutral 3D format. For visualization of product data in the engineering field—regardless of all respective CAD formats—a plethora of 3D formats is available. Among these are disclosed or standardized formats like PDF from Adobe, JT and also X3D, Collada and STEP. The choice of a format has many implications like availability for using the data and the resulting follow-up costs. The discipline of lifecycle visualization aims to tackle problems arising from particular choices. This chapter presents an overview of the industrial challenge, technical background and standardization, typical applications, and evaluation and testing in the field of engineering visualization with neutral 3D formats. The chapter is ends with an approach for assessing 3D formats and with examples from industrial practice in various fields.

In Chap. 12 Goran Šagi, Zoran Lulić, and Ivan Mahalec discuss the concept of Reverse Engineering (RE). A very time-consuming aspect of creating 3D virtual models is the generation of geometric models of objects, in particular when the virtual model is derived from a physical version of the object. A variety of commercially available technologies can be used to digitize very small objects, but also very large. The process of 3D digitizing basically consists of a sensing phase followed by a rebuild phase. Leading CAD software packages include special modules for performing the different tasks in these phases. Many commercial vendors offer sensors, software and/or completely integrated systems. Reverse engineering focuses not only on the reconstruction of the shape and fit, but also on the reconstruction of physical properties of materials and manufacturing processes. Reverse engineering methods are applied in many different areas, ranging from mechanical engineering, architecture, cultural heritage preservation, terrain capture, astronomy, entertainment industry to medicine and dentistry. In the chapter the authors discuss the need for RE in the context of Concurrent Engineering. They introduce various methods and tools that exist today. The chapter ends with applications in different areas like the automotive industry, the railway industry, machinery, architecture, archaeology, and medicine, including ethical and legal issues.

In Chap. 13 Roberto Riascos, Laurent Levy, Josip Stjepandić, and Arnulf Fröhlich discuss the concept of DMU as a validation instrument for evaluating a project's progress by spatially visualizing the current status of the virtual product. Product development in the mobility industry is characterized by an extreme time-to-market, high product complexity and variability, high cost pressure and many geographically dispersed stakeholders. DMU takes over what today's CAD and PDM systems alone are not capable of. The authors extent the discussion to Functional DMU (FDMU), which takes the function of the product into account and enables system users to experience these functions. DMU offers a straightforward visual human interface for control. DMU creation, calculation and processes can be automated well, so that the spatial test (collision check, assembly check) can be performed for all conceivable product variants in batch during the night. Nevertheless, human intervention is still required for the solution of design conflicts. Although not all current problems have been solved yet in the context of DMU, leading PLM vendors offer powerful tools to support the DMU process. Due to its central role in the development process DMU is the subject of intensive research and development for speeding up the development process and to increase accuracy.

In Chap. 14 Josip Stjepandić, Egon Ostrosi, Alain-Jérôme Fougères, and Martin Kurth give an overview of tools and methods to support modular design and management. Modularity has emerged as a relevant way to meet customer requirements with a wide range of variety and customisation of products, from unique to standard ones. The modularity area is becoming increasingly multidisciplinary, presupposing holistic and articulated CE approaches. Modularity, as discussed in the chapter, can bridge the gap between technical aspects and business aspects. Achieving modularity requires both design for modularity and management of modularity. Methods for supporting modular design are evaluated in relationship with technologies and tools for modular design. Although from a holistic point of view there is still much to be desired for achieving system-wide solutions for modular design processes and platform-based product development, the current trend is toward usage and integration of different technologies such as advanced CAD systems, product configurators, agent-based systems and PDM systems. CE approaches are needed for the development of intelligent models and intelligent tools as well as the development of intelligent modular products. The chapter presents different scenarios from real practice for configuring tools for modular design and management.

In Chap. 15 Cees Bil introduces the topic of Multidisciplinary Design Optimization (MDO). MDO has been a field of research for 25 years. It refers to the formulation of the design problem in mathematical models and applying optimization techniques to find the minimum or maximum of a predefined objective function, possibly subject to a set of constraints. MDO has become an important tool in CE, with the ability to handle many design variables across various disciplines. Advances in computer technologies and software engineering have facilitated the practical application of MDO in industry, including aerospace, automotive, shipbuilding, etc. However, active research and development in MDO continues, because there is growing awareness of the criticality of the creative input of the human designer to the design process. For MDO to be effective in the design of modern complex systems it must also incorporate non-technical disciplines, such as finance, environment, operational support, etc. It remains a challenge to model them with adequate fidelity, since simulations and analytical models have imbedded assumptions, inaccuracies and approximations. The chapter gives an introduction to Multidisciplinary Design Optimization with an historical review, a discussion on available numerical optimization methods each with their specific features, various MDO architectures and decompositions and two case studies of successful applications of MDO.

In Chap. 16 Lutz Lämmer and Mirko Theiss address the concept and implementation of Product Data Management (PLM). PLM is widely understood as concept for the creation, storage, and retrieval of data, information, and, ideally, knowledge throughout the lifecycle of a product from its conceptualization or inception to its disposal or recovery. PLM is seen in industry as one of the core concepts to fulfill a number of business requirements in the manufacturing industry with respect to completeness, high transparency, rapid accessibility and high visibility of all product data during a product's lifecycle. Those requirements are related to financial aspects, to the product itself, and to regulatory aspects. PLM is implemented by deploying IT systems like PDM systems and induces a high level of interoperability of related applications. With PLM industrial companies attempt to gain advantages in shorter cycles, lower costs, and better quality by avoiding errors and misunderstanding. After reviewing basic concepts and building blocks of PLM the authors provide empirical evidence of implementation scenarios and use case studies for different PLM solutions. Evaluations of applications in automotive, aerospace and consumer electronic industries are presented, focused on engineering design, change management, simulation data management integration and communication with partners.

In Chap. 17 Georg Rock, Karsten Theis, and Patrick Wischnewski present the topic of Variability Management. The global market, different and changing environmental laws, the customer wish for individualization, time-to-market, product costs, and the pressure on manufacturers to discover new product niches, to name only a few variability drivers, result in an ever-increasing number of product variants in nearly all engineering disciplines like in car manufacturing. Mastering the related increasing product complexity throughout the whole product lifecycle is and remains one of the key advantages in competition for the future. Currently for a manufacturer, as for any other discipline, it is essential to invest in an efficient and effective variability handling machinery able to cope with the arising challenges. Not only the task to invent, develop, introduce and manage new variants is important but also to decide which variant to develop, which one to remove and which one to not develop at all. The consequences of such decisions with respect to product-line variability have to be computed based on formalized bases in such a way that an optimized product variability can assure on the one hand customer satisfaction and on the other hand cost reduction within the variability-related engineering processes. The chapter presents current research in the field of product variability configuration, analysis and visualisation. It presents solution sketches based on formal logic illustrated by some real world examples.

In Chap. 18 Josip Stjepandić, Harald Liese, and Amy J.C. Trappey discuss the important topic of Intellectual Property Right (IPR). With the growth of the knowledge-based economy, IPR is recognized as a key factor to develop and protect strategic competitiveness and innovation of an enterprise. The increasing degree of collaboration in global relationships, ubiquitous digital communication techniques as well as tough competition has lead to an increasing importance of intellectual property protection (IPP) for enterprises. Offences of the law and ethical principles as well as the rising crime through the misuse of modern ICT technologies ("Cyber Crime") pose a significant problem for each market leader. Intellectual property is stored in product data like in modern parametric and feature-based 3D-CAD systems. Because it is very easy to exchange huge amounts of product data between an enterprise and its supplier network there is an enormous threat that intellectual property could fall into the wrong hands and badly jeopardize the existence of the related company. The chapter contains a discussion on the need for action in supply chain networks and attempts by research and development as well as best practices in industry for various aspects of IPP in the context of concurrent engineering.

1.6.3 Part 3: Applications of Methods and Tools for CE

In Chap. 19 Dietmar Trippner, Stefan Rude, and Andreas Schreiber discuss the methods of model based product development, which are well-recognized and wide spread not only in the automotive industry but also increasingly in the aerospace industry and their suppliers. Current challenges of these industries, like light-weight design, electro mobility, modern mobility concepts, plus those caused by rising product complexity, bring this concept to its limits. An overall approach is progressively requested, which is able to continuously integrate requirements, functions, logic and physical product descriptions (RFLP). This should be possible not only for mechanical aspects but also for electronics and software development. The approach of system engineering addresses the continuous availability and linkage of product information. This concept, which has been well known in the aerospace industry for a long time, is only recently used in automotive industry. An example is the use of integrated development environments. Nonetheless, the realization of this concept in an automotive company is definitely a challenge. Examples for these problems are differently coined, like detailed requirements (client requirements versus requirements to a complete vehicle and to components properties), consideration of configuration, validity and maturity, complexity management (complete vehicle to component, vertical integration, plus integration of early concept phases over development, verification, clearance to the production start-up, horizontal integration) and multi-disciplinarity (mechanics with calculation, electronics and software). The realization of systems engineering does not only create high demands to the design of process-IT (authoring systems, TDM and PDM), but also has to consider organizational aspects. Frequent acquisitions for IT system vendors, especially in the CAD /PLM/CAE market, as well as the selection of systems for functional and economical aspects lead to increased requirements concerning open interfaces. In the chapter, findings and experiences from the introduction of systems engineering for automotive processes are described. Effects on the process IT architecture are outlined. "Lessons learned" and necessary changes in process-IT, in the form of selected examples and solution alternatives, are discussed.

In Chap. 20 Richard Curran, Xiaojia Zhao, and Wim J.C. Verhagen address the application of CE in aviation by explaining the relationship of CE and integrated aircraft design. Driven by rapidly increasing air traffic worldwide, the main players in the aviation market have initiated many new development projects in the past decade with increasing size and complexity. Such projects presuppose an integrated and advanced design process that is able to ensure the concurrent synthesis of many life cycle performance drivers within a complex and collaborative aviation enterprise. Furthermore, aspects of the product's life cycle, like the overall cost performance and the ability of new system integration, were adopted at an early stage in the design process. This gives the PLM its crucial importance managing the extended enterprise over a long period of time. Consequently, the implementation of CE in the lifecycle of an aircraft and systems in general is illustrated in the context of the high structural complexity and long product lifecycle compared with other application areas of CE. Further challenges related to process parallelization and multidisciplinary design, involving the exchange of knowledge and information throughout the design process, are covered. Supporting techniques like DMU along with practical case studies are presented with several examples to illustrate the implementation of CE and integrated aircraft design in real life. Expected future developments with respect to concurrent engineering, as applied to aviation, conclude this chapter.

In Chap. 21 Alfred Katzenbach discusses the use of information technologies for product development in the automotive industry. The automotive industry is one of the most advanced industries in this respect. Product variety and complexity has grown dramatically over the last decades, making the use of information technologies as presented in part 3 indispensable. Automotive engineering companies are looking continuously for new ways of economic growth. Trends show that this is often done by expansion of existing markets as well as entering new markets, providing niche products and increasing productivity. This affects significantly the continuous development of processes and IT solutions. Legacy Systems have to be integrated with modern solutions. Service-oriented architectures and semantic nets will lead to new system landscapes. However, this change is not only a technical one. It is also an organizational paradigm shift, which has to be handled carefully. To establish an international, multi-company CE process, common understanding of processes and business objects is required. The most efficient way to do this is standardization. The chapter presents the "Code of PLM Openness" (CPO), which helps to find a common definition that leads to a better understanding of system integration and usage of standards. Two Standards play a significant role: ISO

10303 (STEP) with its new application protocol 242, which combines the known protocols for automotive and aerospace including model-based systems engineering and ISO 14306 (JT) for DMU and geometrical collaboration. By enhancing CAD systems a knowledge-based engineering approach will become reality.

In Chap. 22 Jožef Duhovnik and Jože Tavčar discuss the application of CE to machinery. This application has to consider the type of production (individual, serial), product complexity and level of design. Product development involves four characteristic levels of design that requires specific activities. The characteristic design levels require definitions of the activities for providing the necessary software and other support for all phases of the design process. The following four levels of the design process have become established in the professional literature: original, innovative, variation and adaptive. Systematic analyses in various companies of product development processes (PDP), workflows, data and project management has shown that specific criteria have to be fulfilled for CE to be managed well. It is very important to consider the involvement of customers and suppliers, communication, team formation, process definition, organisation, and information system to fulfil minimum threshold criteria as well as the phase of development. The quality of communication and team formation, for example, primarily affects the conceptual phase. An information system is useful predominantly in the second half of the design process. In the second part of the chapter reference models for CE methods are presented for product development in individual production (CE–DIP), in serial production of modules or elements (CE–DSPME) and in the manufacture of mass products (CE–DMMP) with an example from household appliances. The reference models for CE methods map product development phases and CE criteria for each type of production and have to be used together with case studies. They help to recognise strong and weak points of a CE application and show a way to improve processes and supporting CE methods.

In Chap. 23 Kazuo Hiekata, Matthias Grau discuss the sharing of information in the shipbuilding industry. The shipbuilding process generally consists of concept and preliminary design, basic design, detailed design, production design and production. Design information is generated in each phase to shape products and operations in the shipyard. For each process the design activities are carried out with a high level of concurrency supported by various computer software systems, though quality of products and efficiency of the concurrent development process highly depend on experiences and insights of skilled experts. Detailed design information is difficult to share, while design conflicts are solved in a common effort by design engineers in downstream design stages. Data sharing across design sections and simulation of the construction process to predict time and cost are the key factors for CE in shipbuilding industry. The CE process in shipbuilding will become more and more accurate and efficient along with accumulation of design knowledge and simulation results. The chapter gives insight into the different phases of the shipbuilding product creation process and demonstrates practical usage through typical, comprehensive use cases from design and manufacturing. It also presents some expected future directions for CE in shipbuilding.

In Chap. 24 Chun-Hsien Chen, Li Pheng Khoo, and Nai-Feng Chen discuss product design and development (PDD) in the context of consumer goods. PDD has shifted its focus from addressing functional and technological issues to user-centric and consumer-oriented concerns in recent years. More specifically, the experiential aspect of design has taken a crucial role in creating more consumer-focused products. Often, customer research or user-involvement studies are conducted to acquire the necessary knowledge and gain insight into experiential requirements of users. Unlike functional requirements, customer experiences are usually more tacit, latent and complex. More attention for acquiring these experiences is needed. In the chapter a prototype Context-based Multi-Sensory Experience System (CMSES) with a Scenario Co-build Strategy (SCS) is proposed to facilitate user experience acquisition in designing consumer goods. A three-stage case study is described to illustrate the proposed prototype system. The potential of the proposed approach in the context of CE and collaborative product development (CPD) is discussed.

In Chap. 25 Osiris Canciglieri Junior, Maria Lucia Miyake Okumura, and Robert Ian Marr Young discuss the importance of CE for the design of medical equipment. Design of medical equipment requires a multidisciplinary approach. In the chapter a multidisciplinary environment for Integrated Product Development Process (IPDP) of medical equipment is presented. The authors address the requirements of a health professional user as well as patient's needs. The medical equipment lifecycle has been identified and contextualized. The importance of CE in the IPDP of medical equipment has been shown and propositions have been presented for the insertion of software tools that support various product development phases. A discussion is included on the use of CE and IPDP oriented towards medical equipment conception and development, perspectives of engineering modular development and interfaces between Health and Engineering information areas for increasing technical, clinical and economic quality.

In Chap. 26 Amy J.C. Trappey, Charles V. Trappey, Jerry J.R. Ou, C.T. Hsiao, Kevin W.P. Chen, and Penny H.Y. Liu introduce an approach for assessing the economic input-output lifecycle (EIO-LCA) and a location quotient (LQ) for measuring regional carbon footprints using local environmental and industrial data. Countries and government regions are promoting renewable energy to effectively reduce carbon emissions. However, the carbon footprint of a given industry in a specific region is hard to measure and the long-term effect of an untested green policy for carbon reduction is difficult to predict. The results of the proposed approach enable government policy makers to accurately formulate policies that target critical contributors while simulating the economic impact using system dynamics (SD) modeling. In a case study, policy scenarios are simulated to evaluate the time-varying impacts of proposed green transportation strategies for Taiwan's low carbon island (Penghu Island) pilot project. The methodology provides a generalized tool for green energy policy assessment.

In Chap. 27 Alain Biahmou discusses the concept of sustainable mobility as a field of application of Concurrent Engineering. In particular, the electrical power train of road vehicles has an increasingly significant role. Besides delivering benefits in air and noise pollution, it encompasses huge challenges in practical usability, reliability and total costs of ownership combined with novel models of exploitation. The design of electric vehicles requires bringing components from different domains together to integrate them in the overall vehicle concept. The domains involved utilize their own specific methods, processes as well as software tools in order to create partial models of an overall system. This leads to dependencies between several disciplines and, therefore, to the need to track the impact of model interactions to avoid data inconsistency as well as design errors. The focus of the chapter lies on the project "Process Chain Battery Module" that has been conducted at EDAG Engineering AG to capture the challenges related to the electrical battery when designing electric vehicles. Thermal management, which is one of the critical challenges to be tackled in the area of electro mobility, is discussed and solution approaches are presented. Requirements are defined and linked with functional analysis as well as geometrical, behavioral and FEM models. Thus, changes can be traced from each partial model back to the initial requirements. Interface management between the domains and partial models is realized to enable an analysis of the entire vehicle. Complex simulations are performed in a very early stage of development to determine the range of an e-vehicle model (EDAG Light Car).

1.6.4 Part 4: Current Challenges

Chapter 28 summarizes the research and practical challenges that have been presented in the book. It will pay attention to the need for new theories as well as the need for new methods and tools. It will also pay attention to the current barriers for improvement. The challenges have been identified by analysis of all chapters according to a common framework that encompasses the socio-technical dimensions of CE. Much work is still ahead to improve the concurrent way of working in all phases of a product development process. The needs for CE methods and tools are, however, very different for the various phases of the development process. While much effort is still needed to improve interoperability and transparency for smooth data exchange in later phases, support of collaboration between people, especially in the earlier phases, is still subject to improvement. A CE approach in research, bringing together different disciplines, is needed to generate suitable tools and methods for supporting the earlier phases, characterized by large uncertainty and risks.

1.7 Contributors of the Book

The editors have selected and invited contributors based on their recent contribution to CE conferences. Additionally, industry experts have been invited to contribute. The editors are grateful to all contributors for their excellent work.

References

1. Stjepandić J, Rock G, Bil C (2013) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. In: Proceedings of the 19th ISPE international conference on concurrent engineering, Springer, London
2. Bil C, Mo J, Stjepandić J (2013) In: Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam
3. Cha J, Chou SY, Stjepandić J, Curran R, Xu W (2014) Moving integrated product development to service clouds in the global economy. In: Proceedings of the 21st ISPE Inc. international conference on concurrent engineering, IOS Press, Amsterdam, 8–11 Sep 2014
4. Backhouse CJ, Brookes NJ (1996) Concurrent engineering: what's working where. Gower Publishing, Aldershot
5. Prasad B (1997) Concurrent engineering fundamentals: integrated product and process organization. Prentice Hall, Upper Saddle River
6. Hartley JR (1998) Concurrent engineering: shortening lead times, raising quality, and lowering costs. Taylor & Francis, New York

Part I
Foundations

Chapter 2
The System of Concurrent Engineering

Nel Wognum and Jacques Trienekens

Abstract Concurrent engineering (CE) has been a major theme in the 80s and 90s of the previous century in research and practice. Its main aim is to reduce time-to-market, improve quality and reduce costs by taking into account downstream requirements and constraints already in the design phase. While starting with a design-manufacturing alignment, gradually the CE way of thinking has been ex-tended to incorporate more lifecycle functions together with a stronger focus on and involvement of both customers and suppliers. Application of CE in practice has led to remarkable cost savings, time reduction and quality improvement. However, many failures have been reported too. Often, the complex system of CE has not been sufficiently well understood, in particular because the system that is needed to market, produce, sell, and maintain the new product, the so-called production system, has not been considered sufficiently. The particular properties of the production system that is needed to really make the new product a success need to be understood well, because they heavily influence the CE process. In this chapter a history of CE is sketched as well as its major achievements and challenges. The essentials of the system of CE are described together with the system that is designed by it: the production system. The production system, as defined in this chapter, is an encompassing system, because it also comprises functions like marketing, sales, production, and maintenance. The interaction between the two systems needs to be taken into account in all CE processes in any application domain. The chapter ends with examples of the food application area. The variety of the system of CE, in terms of different innovation efforts, is illustrated. Some important properties of the result of a CE process, a food production system, are discussed, in particular a food supply chain and its coordination for quality.

N. Wognum (✉) · J. Trienekens
Social Sciences Department, Management Studies Group, Wageningen University, Hollandseweg 1, 6706KN Wageningen, The Netherlands
e-mail: wognumnel@gmail.com

J. Trienekens
e-mail: jacques.trienekens@wur.nl

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_2

Keywords Concurrent engineering · Collaborative engineering · Early supplier involvement · Collaborative/open innovation · CE system · Production system

2.1 Introduction

In the '80s of the past century, the term concurrent engineering(CE) was coined to indicate a way of working in product development and design to meet consumer demands in shorter time, with fewer errors, and lower costs. CE was meant to improve industry's competitiveness especially in the West to catch up with the advantage gained by Japanese companies like Toyota.

The essence of CE has been the concurrent execution of design processes with the design of downstream processes, in particular manufacturing. Teams of multiple functions and disciplines were formed to discuss design proposals from different, multidisciplinary, point of views. In these teams, design disciplines, manufacturing and assembly, marketing and purchasing were often represented.

Several new terms were used to indicate more specific approaches of CE, like Design for Manufacturing and Design for Assembly. Since many design and development processes also required the involvement of external technology providers the term Collaborative Engineering has also been used to indicate the concurrent way of working.

Later, more downstream processes became involved in CE, like service and disposal. The necessity to incorporate the customer early in the design process was also recognized in the approach called Open Innovation, in which consumers, customers, suppliers, and OEMs collaborate to identify potentially successful product ideas. In this way waste in terms of time and cost is reduced considerably by the upfront matching of insights of important stakeholders.

All these approaches basically center on boundary-spanning processes. Although the term CE is hardly used anymore, process thinking and boundary-spanning processes have gained more and more attention. Current business requires collaboration between companies, like networks and supply chains, to maintain or improve their market position. Information technology plays a large role in supporting information sharing and aligning people and companies.

Application of CE in practice, in whatever disguise like early supplier involvement and design network, has led to many improvements and time and cost reduction. However, achieving a well-performing CE system is not at all an easy process. It may take years of gradual change and building experience. Moreover, top management needs to have a clear vision. A systems approach is helpful to identify the essential elements that are involved in the CE process as well as their interrelationships. In addition, a clear view on the production system that needs to market, produce, and maintain the new product is necessary.

In this chapter, essential properties of the CE approach are discussed. In Sect. 2.2, the history of CE is briefly discussed together with its achievement as recorded in the literature. In Sect. 2.3, the system of CE is described with its essential elements and relationships. A framework with steps for framing a CE process is presented. A CE system is closely linked to the production system that is the result of or needs to be taken into account in a CE process. The production system is a complex system that needs to be designed or adapted well to make its product (or service) a success. In Sect. 2.4, some examples of the system of CE in the food industry are presented. The framework with steps from Sect. 2.3 is used to frame the relevant innovation processes and indicate the relevant production system that needs to be taken into account. The paper ends with a summary and future challenges.

2.2 History of CE

In this chapter, a brief history is sketched of the main developments in CE in the past decades. First, CE is briefly described, including Early Supplier Involvement. Second, Collaborative Engineering is described, followed by Collaborative Innovation.

2.2.1 Concurrent Engineering

In the 80s, companies were forced to change the product development process from the traditional 'over the wall' approach to more integrated ways of working to beat growing competition, react to reduced product life cycles, and meet changing market and customer demands [1]. They needed to be able to develop new products, which were cheaper, delivered faster and provided a greater functionality [2]. CE was considered to offer a solution to the problems encountered.

CE has had tremendous attention in the literature since Winner et al. [3] in the DoD Institute of Defense Analysis coined the term. The concept has resulted from a USA Defense Advanced Research Projects Agency (DARPA) initiative to improve the product development process. The first definition of CE was [3]:

> Concurrent Engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support. This approach is intended to cause the developers from the outset to consider all elements of the product life cycle from conception to disposal, including quality cost, schedule, and user requirements.

This definition stresses the parallel, concurrent, execution of product and process design activities by integrating multiple design disciplines and upstream and downstream functions involved in the lifecycle of a product. Many studies have been devoted since then on further defining this concept. CE is known under various different names, like Simultaneous Engineering, Concurrent Product Development, and Integrated Product Development with definitions slightly different from the one above (see e.g., [4, 5]).

CE has three basic elements: early involvement of participants, the team approach, and the simultaneous work on different phases of product development [6]. CE teams typically consist of the functions marketing, product engineering, process engineering, manufacturing planning, and sourcing activities. The principle focus was the integration of and alignment between design and manufacturing functions, while taking into account consumer demands and supplier capabilities.

Conflicts easily arise within cross-functional CE teams because of different interpretations leading to confusion and lack of understanding. Each team member focuses on different aspects, like marketing on usability, engineering on functionality, production on manufacturability, and purchasing on affordability [7]. In such situations, communication needs to be predominantly personal and involve face-to-face contact.

The early involvement of relevant stakeholders in the design and development process enables exchange of preliminary information. Such information exchange may reduce the number of engineering change orders, which are often the reason for delay in product development projects (see e.g., [8]). Strategies for the exchange of preliminary information exchange may differ with the level of downstream uncertainty and costs of process idleness [9].

To support collaboration in teams and facilitate information exchange and use, many attempts have been made to develop engineering knowledge and collaboration tools (see e.g., [10]). They are, however, still poorly developed [11]. As reported by Lu et al. [11] based on a document from the EU-funded FP6 project VIVACE [12] 26 % of project meetings in Airbus involve international partners, more than 400 one-day trips were taken by Airbus engineers to collaborate with other project members on a daily basis, while they also spent an average of 49 % of their daily activities in meetings and discussions with stakeholders. It can be said that engineering has become a highly collaborative activity in today's industry.

2.2.2 Collaborative Engineering—CE*

Gradually, the number of stakeholders that needed to be involved in the design process increased. In particular, the marketing and purchasing functions and downstream functions like service and asset recovery have been involved early in the design process. Because products are used and need to be disposed eventually, environmental concerns have also added to product design and development complexity (see e.g., [13]).

The desire for incorporating multiple lifecycle considerations requires tight integration of multi-disciplinary knowledge and collaboration between engineers across various cultural, disciplinary, geographic and temporal boundaries [11]. Todd [14] has defined collaboration as the process of multiple people working together to achieve a greater goal than is possible for any individual to accomplish alone.

Putting the emphasis on collaboration has led to the term Collaborative Engineering (CE*) with the following definition [15]:

> Collaborative Engineering is a systematic approach to control lifecycle cost, product quality, and time to market during product development by concurrently developing products and their related processes with response to customer expectations, where decision making ensures input and evaluation by all lifecycle functions and disciplines, including suppliers, and information technology is applied to support information exchange where necessary.

In addition to involving the purchasing function early in the design process the supplier itself has become a team member. Together with the buyer the parts and materials supply as well as the required logistics are taken into account as early as possible [7]. In addition, the supplier could take responsibility for (parts of) the development process or be involved in different phases, like concept design, engineering, or process engineering (see e.g., [16]).

Early supplier involvement (ESI) as part of CE and CE* has received much attention from researchers and practitioners at the end of the 90s and early 2000s. A literature review by McIvor and Humphreys [16] revealed that despite the potential benefits of ESI negative impact of various factors might exist, like technology uncertainty, low levels of trust between the buyer and supplier, poor communication and co-ordination mechanisms. These factors are similar to those mentioned often also in the context of CE and CE*. Development and monitoring of collaborative relationships are critical for preventing problems with supplier performance [17].

Because fundamental knowledge about human collaboration and its underlying sciences is lacking, a CIRP community has attempted to start a new human-centered engineering discipline by developing a first step of a socio-technical theory of collaborative engineering [11]. This theory builds on various theories from collaboration science, like organizational behavior, social psychology, social choice and decision science. However, many challenges still exist for further developing the socio-technical theory of collaborative engineering.

2.2.3 Collaborative Innovation—CI

Research and Development in large companies used to be internal in the past decades. Many R&D project, however, have led to results that appeared not to be useful for the respective companies leading to waste in terms of time, money and missed market opportunities. However, some of those results, although not valuable for the company itself, turned into valuable spin-off companies [18]. To limit waste and increase the success rate of technology projects, a new business model gradually emerged from Closed Innovation (with extensive control) into Open Innovation.

Open Innovation requires collaboration between a firm and external sources of knowledge, like technology providers, start-ups, small enterprises, consumer organizations, etc. External knowledge increases the potential number of innovations, while also external parties can exploit internal knowledge. Procter & Gamble (P&G), for example, changed the concept of R&D into Connect and Develop (C&D) [19] to indicate the necessity to open up its knowledge and admit external knowledge to keep up and improve its competitiveness. Its experience with CE and CE* models allowed P&G to transit to the new model in reasonable time.

Gradually, the concept is also adopted in more traditional and mature industries like the food industry [20]. As argued by Sarkar and Costa, since the number of actors is large and no one actor alone can meet all, often contradictory, requirements of customers, consumers and legislation bodies, open innovation should be common practice. However, empirical evidence is still anecdotal to date, although the necessity and need for open innovation is gradually recognized.

Vanhaverbeke [21] argues that the open innovation business model should be based on integration of theoretical frameworks, like value chain analysis, transaction-costs theory, rational view of the firm, and the resource-based view (RBV). In addition, governance of innovation networks, on internal, firm and external level, needs to be studied. Networks or supply chains that will eventually produce the product need to be designed also with appropriate governance (see also Sect. 2.4).

2.2.4 CE Success and Failure

CE, CE*, ESI, and CI are approaches requiring collaboration within and across organizational borders. These approaches present complex problems that require a socio-technical approach in which both the technical and social systems and their interaction are taken into account. Koufteros et al. [6] have found that firms that have adopted CE practices report better performance in product innovation and quality, while they are also able to charge premium prices. A firm's internal context is important for facilitating cross-functional integration. Once achieved, external integration is sought for with customers and suppliers to coordinate activities across the value chain. Information technology is an enabler for this way of working. Many success stories can be told with reductions in product development time of 50–70 % (see e.g., [1]), but also many failures.

In the early 90s, a survey of Swedish manufacturing firms showed that Swedish firms had a broad awareness of the importance of product development [1]. Various names have been given to the CE way of working, with integrated product development as the one most widely used. Reducing lead-time was considered the most important goal for CE, followed by customization of products. The dominant element of CE for achieving lead-time reduction is the use of multifunctional project teams, sometimes including customers as well as suppliers, especially in companies successful in reducing lead time (about 50 %). However, such teams are not sufficient for success, because also companies not successful in reducing lead-time (about 23 %) appeared to be using them. Additional methods are needed, like Quality Function Deployment (QFD) and Failure Mode and Effect Analysis (FMEA), which were used by the most successful companies as well and typically in aerospace. In addition, CAD/CAM integration also was more widespread in such companies.

In another study in British industry Ainscough and Yasdani [22] have found that CE was not uniformly spread among British industry sectors. Of the large companies, 100 % claimed to practice CE, with only 63 % of medium-size companies with (101–500 people) and 50 % of the smaller companies. Medium and large companies heavily relied upon formal product development processes, multifunctional teams, tools and techniques, information technologies, and project management activities for executing CE. The functional structure is not suited for executing complex projects like CE projects making various integration mechanisms necessary, although small companies do not seem to need extensive integration mechanisms, because people are closer together.

Implementation of organizational structures needed for executing CE, CE*, ESI, and CI projects, including complex information technologies needed to handle and share the large amounts of information involved in such projects, requires extensive organizational change. As with all major changes, observable also with implementation of ERP systems and other integrated systems, various factors play a role in making such changes a success or failure. McIvor and Humphreys [16] have listed factors that play a role in adopting ESI, which are not much different from the factors mentioned for other major organizational changes.

In the next section the complex system of CE is further explored together with the encompassing production system that is both the result of the CE process and a constraint for it.

2.3 The System of CE

As has become clear in the historical development sketched above, CE has increasingly become a very complex system with many players, locations, information systems, methods and tools that have many different and layered relationships. In addition, as CE is a process of product development and design with the aim to make the product a success, the specific properties of the application domain have to be taken into account. Although the principles of CE are general, they need to be made specific for the different application domains. For example, knowledge of the essential properties of products in specific application domains is needed. These properties influence the production processes that need to be designed together with the design of the product. For example, the short lifecycle of engineering products like cell phones limit the creation of production systems like new companies or supply chains, unless a totally new product family is started. Food products in general have rather long lifecycles, which may justify the creation of new food supply chains. On the other hand the short life of food products, due to perishability, puts heavy constraints on storage facilities, while engineering products may allow unlimited storage during and after production. The production processes, consequently, have different properties to be taken into account during a CE process. In addition, user preferences of products in different application domains may also be rather different. As already indicated above, market knowledge is essential for any CE process to succeed.

While the previous sections have explored developments in CE as well as the essential actors, information systems, methods and tools and their mutual relationships, the production system that needs to be taken into account has only briefly been addressed. Below, the system of CE with its elements and relationships is further explored in connection to its resulting production system. The next section introduced the concept of an organisational system as a basis for the CE system and the production system to be discussed in the subsequent section.

2.3.1 Organisational System

A system, more specifically an organisational system, is inherently a socio-technical system in which technology and organisation need to be aligned with each other to achieve the envisioned organisational goals. In a socio-technical system many different disciplines need to collaborate, while taking into account the social context in which collaboration takes place.

An organisational system can be defined as a purposeful whole in which people perform processes with the help of means, like methods and tools, to satisfy certain needs in the environment of the organisation [23]. These processes are essential for achieving the organisational goals. They transform inputs, like material and/or information, into outputs in the form of products and/or services needed in the environment. There are different types of co-existing processes, like strategic management processes, adaptation or improvement processes, and operational processes, including operational management, primary, and support processes, like the processes of human resource management, maintenance and education. The elements of the system, which are the people, processes, and means, are tightened together by organisational arrangements. These are all formal and informal structural and cultural relationships between the elements. Formal structural relationships reflect the hierarchy, tasks, and procedures as laid down in quality handbooks. Informal structural relationships consist, for example, of the routines that have been adopted by the workforce. Formal cultural relationships are, for example, the norms that underlie the manners and moral of the people in an organization. Informal cultural relationships are manifested in the way people communicate with each other during meetings or coffee breaks.

An architectural view of an organisation with a focus on information flows is depicted in Fig. 2.1 [24]. The figure shows all elements and relationships mentioned above. Most importantly, the figure emphasizes that an organisation consists of activities (making up processes) that are performed by resources (people and means), while organisational arrangements relate activities to resources in the form of procedures, hierarchy, tasks, roles, norms, etc. Activities can be divided into transformation activities and communication activities. Transformation activities transform input information into output information. They also transform input material into output material, but this is not shown in the figure. For example, in a design process product requirements are transformed into a conceptual design; in an

Fig. 2.1 Organisational system with a focus on information

order processing process information of the products status is transformed when the order is processed in a production system. Communication activities transfer information from one activity to another and, as such, between the people that perform the activities. The document flow represents the formal communication in an organisation, while informal communication is more difficult to grasp.

The processes performed in an organisational system determine its behaviour. Processes are expected to proceed as designed, i.e., as laid down in process schemes and handbooks in ISO certified companies. However, this is often not entirely the case, not only because of unexpected disturbances like broken machines or insufficient supplies, but also because of the culture, politics, power, and other aspects involved in collaboration between people [25].

Organisational arrangements as already indicated above represent the structure and culture of an organisation. Part of these arrangements are the normative relationships between elements in the organisation, like organisational hierarchy, reporting relationships, process structure, infrastructure, team structures, procedures and routines, as well as the values, rules and norms that constitute a relatively coherent and consistent set of beliefs and prescriptions that govern the behaviour of people [26]. Normative relationships constrain and channel human behaviour in an organisation. Normative relationships are not static, but are subject to change over time.

Actual behaviour often differs from behaviour intended in the normative part of organisational behaviour. Actual behaviour not only depends on individual human characteristics, but also on relationships and interactions between people who bring their own technical and social knowledge and experiences to the organisation. Commitment, attitude, sentiments, conflicts, autonomous activities are examples of characteristics that influence organisational behaviour. Actual behavior that positively influences organisational performance often is memorized and shaped into normative relationships.

The interaction between normative relationships and actual behaviour is called duality [27]. Actual behaviour may shape the normative relationships, while normative relationships shape behaviour. Actual behaviour is often called social-dynamics.

A system is more than the sum of its elements [28]. The behaviour of a system as a whole cannot be found in any of its elements. A system view, therefore, is a holistic view. It is possible to describe and analyse parts or aspects of a system, but without taking into account their relationships with other parts or aspects, conclusions may not be very reliable. A system view offers an analytic way to focus analysis on a coherent part of the world.

Application of the system view to a real-world problem like CE requires that system borders are determined. Examples of systems to be analysed are the manufacturing process [29], the R&D process [30] or the collaboration process between companies [31]. Determining system borders and the relevant system elements, such as the people that need to be involved, starts with the selection of the focus process, like a design process, a collaboration process, an invention process, a marketing process, a purchasing process, etc., which determines the system borders and the environment of this system. The environment can be the environment within the organisation as well as the external context in which the organisation operates. The context depends on which process has been chosen for in-depth study, the so-called process of focus.

The model depicted in Fig. 2.1 has been used to describe an architecture of a virtual organisation to identify essential capabilities needed for mature performance. A virtual organisation is a temporary organisation, often a complex project, in which several companies collaborate to develop a new product. CE processes are often performed in such a virtual organisation. In Fig. 2.2 an architectural view based on Fig. 2.1 is depicted [24]. In this figure, the essential elements are presented. Each organisation in the figure can be described by Fig. 2.1 separately.

A virtual organisation consists of two or more partners, each of which is part of a mother organisation. A virtual organisation is an organisation with its own goals and strategy for which specific activities and organisational arrangements need to be defined. The processes in a virtual organisation are often restricted to coordination (management) processes, support processes and communication processes, while the primary processes, like design and production processes, are often performed in the mother organisations. Part of the primary processes, like idea and concept development, can also be performed in the virtual organisation.

Essential differences may exist between a virtual organisation and the mother organisations involved. These differences can have a large impact on the performance of the virtual organisation [24]:

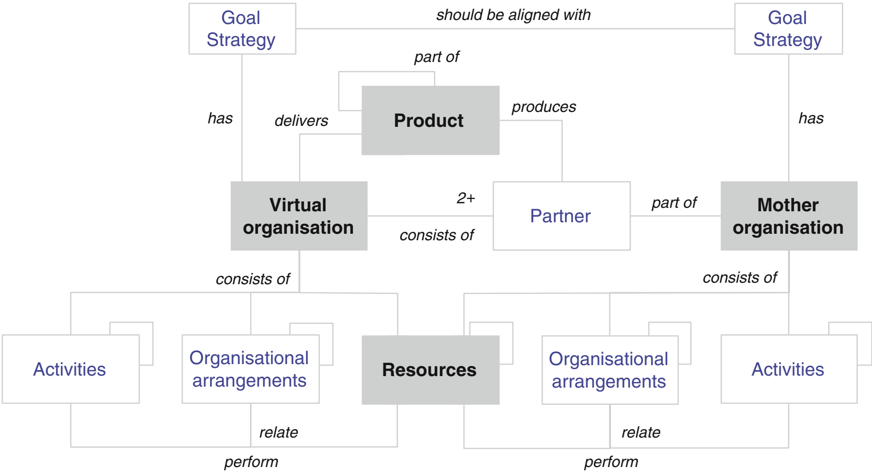


Fig. 2.2 System of a virtual organisation with a focus on information

- The environments (not depicted in Fig. 2.2) of the mother organisations may overlap. For example, in some industries competitors are forced to collaborate (temporarily) in a virtual organisation. Such a situation may hamper collaboration. The transfer of knowledge, e.g., between the people involved may be restricted by national or legal reasons leading to insufficient sharing of the knowledge for performing the processes.
- The resources of a virtual organisation have been assigned by the respective mother organisations to the part(s) that participate(s) in the virtual organisation. Several problems, often reported also in project management literature (see e.g., [32]), may result from this situation. First of all, commitment of the people involved depends on the balance between their normal duties in the mother organisation and their duties in the virtual organisation. Secondly, alignment of competencies, social as well as technical, of people from different companies is essential as well as interoperability of the means (not only technical) involved. Thirdly, the people involved in a virtual organisation have to adapt to the context of the virtual organisation in terms of the social and technical infrastructure defined for the virtual organisation, which is often different from the ones in their mother organisations. However, people will bring working habits, norms and rules from their own company to the virtual organisation. Fourthly, a large part of the activities is still performed in the mother organisation, possibly tipping the scale towards a larger influence of the mother organisation. In summary, large differences between partners, in terms of people, means, organisational arrangements (structure and culture), and goals may negatively influence collaboration in a virtual organisation when not sufficiently recognized.
- The product delivered by the virtual organisation to clients/customers (not in Fig. 2.2) consists of subsystems/parts that are produced in several mother

organisations. This situation poses specific quality demands on product and process information (not in Fig. 2.2) delivered by the partners. For example, difference in terminology and interpretation frameworks (semantics) may disturb communication. In addition, when not sufficiently defined in the virtual organisation beforehand, differences in (documentation) standards and information management facilities may lead to misunderstandings, conflicts and costly delays.

Complex processes, like CE processes and multi-site production processes, as are the focus of this book, are inherently multi-dimensional and multi-level as will be clear from the discussion above. They are multi-dimensional, since they involve different aspects, such as process aspects, people aspects, technological aspects, and organisational aspects. They are multi-level, since they can be specified on the individual level, the group level, the project level, and the organisational level. Moreover, such processes evolve over time, because situations may change and people gradually learn, necessitating changes in goals, activities, and resources. A process approach is necessary to manage collaboration in complex processes, taking into account social dynamics and unexpected events (see Chap. 8).

In the section below the system of CE will be described in general terms together with a process to make this system more specific for a particular application domain. This system description can be used to support analysis of existing CE processes and identify opportunities for improvement. In the subsequent section essential characteristic of the production system, which is the intended output of a CE system, but also constrains this system, will be described.

2.3.2 The System of CE

CE is essentially an innovation system. It is aimed at generating either a totally new product or changes to existing products, which may vary from essential changes to minor variations. As such the CE process influences the production organisation. In the case of new products, a new production organisation may be the result of the CE process, for example a new company with its own supply chain or a new production line in an existing company. During the CE process this new production system is an essential part of the design that is the output of the CE process. In case of adaptions to existing products, the changes that are needed in the existing production system need to be taken into account. These may exist of new tooling, including new tasks and procedures, new materials or parts as input, requiring new suppliers or changes in existing relationships with suppliers. The relationship between the CE system and the production system are depicted in Fig. 2.3.

This view of two co-existing systems may help to frame, study and analyse CE processes in real-life. A step-wise approach, incorporating this view is proposed below. The approach, as all system approaches, is an ‘empty’ framework. For each different situation, a specific description needs to be made with the help of additional theories and knowledge. The steps for framing an existing situation are the following:

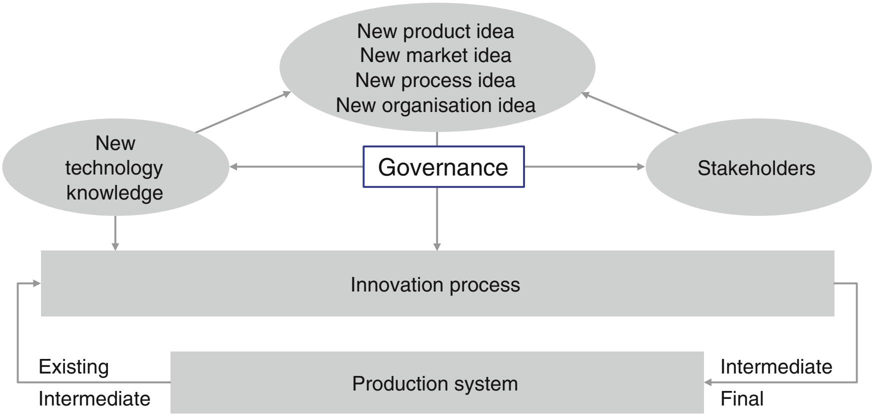


Fig. 2.3 The system of CE

1. Identify the overall goal or vision of the CE system. More and more the goal is encompassing, because not only profitability should be achieved, but also social welfare and reduced impact on the environment. In other words, the CE system should focus on a sustainable output in a sustainable way.
2. Identify the process of focus, i.e., the process that will be the subject of study. A CE process is multi-functional, multi-disciplinary or multi-site, requiring collaboration between different disciplines, functions and roles. For example, the process of focus may be the product development process, the idea generation process, the information management process, etc. The start and end of the process should be determined.
3. Determine the internal and external context of the process. The external context could consist of: departments that influence the process, but are not involved, governmental rules, financial situation, environmental situation, technology providers, etc. If the process is an aspect of a larger whole, like the information management process in product development, the overarching process should be identified. The internal context consists of all departments, organisational levels, or organisations involved including their structural, cultural, and technological properties.
4. Determine the actors, functions or roles that are or should be involved in the process. What does this mean for the departments/organisations that are involved in the process? What humanware is involved in terms of knowledge, expertise and skills?
5. Determine the technologies that are needed/used in the process: hardware and software.

The approach can be used to 'frame' the problem area and identify the specific focus of study. To study this focus process in more detail, additional methods and tools are often needed, for example planning tools. By applying the approach together with these additional methods and tools performance management can be performed, for example, to keep the CE process ‘on track’ technically and socially.

As an example, consider the *open innovation process*. It is in essence an idea generation and development process with prospective partners. It may be aimed at developing a real business case as output. Below, the steps are applied to ‘frame’ an open innovation process.

Goal: Develop a new product idea with specific (sustainable) properties for a specific market (including envisioned investments and pay-back time).

Process: Idea generation and development. Source of ideas may be all employees and partners, like suppliers. The steps depend on the output considered. If the output is a business case, then all steps in the funnel process towards a realistic business case are needed.

Context: The business process that needs to execute the business case, finance available or to be made available for the business case, existing regulations, suppliers not involved in the process, customers not involved in the process, etc. There are also requirements for the business case to satisfy: e.g., added value, short- and long-term results expected, etc.

Actors: representatives of relevant departments, suppliers, customers, client/consumer group. *humanware*: knowledge/expertise needed for the specific domains involved; in this case the knowledge may be varied, because the knowledge needed may depend on the ideas generated and developed; people may be involved later in the process when needed; libraries might also be searched.

Technologies: *hardware*: decision support tools, brown paper tools, etc.; *software*: brainstorm techniques, decision support software, financial software for making the business case, project design systems, etc.

A production system resulting from a CE process is often also a complex system (see Fig. 2.4). It consists of several collaborating companies, in networks or in supply chains. The production processes performed by the actors in the system are influenced and restricted by regulations and quality management systems for which suitable governance mechanisms, with appropriate coordination mechanisms [33], need to be installed to make the system working. In addition, the production processes not only depend on the knowledge, skills, and technology that are available in the system in people, hardware and software, but are also available in the environment through technology providers and knowledge institutes.

It depends on the application area which company is most in touch with the end consumer of the production system or is most responsible for the quality of the end product. The way customer demands are gathered and translated into product requirements may vary between application areas. Knowledge of markets and customers is essential for a CE process to develop a sustainable new product and a sustainable production system satisfying existing regulations and quality management systems. Many information and material flows co-exist.

The system presented in Fig. 2.4 is the production system that plays an essential role in any CE process. By focusing on the product only and not on the encompassing system that is needed to needed to market, produce, and maintain the product with appropriate services, the product might fail.

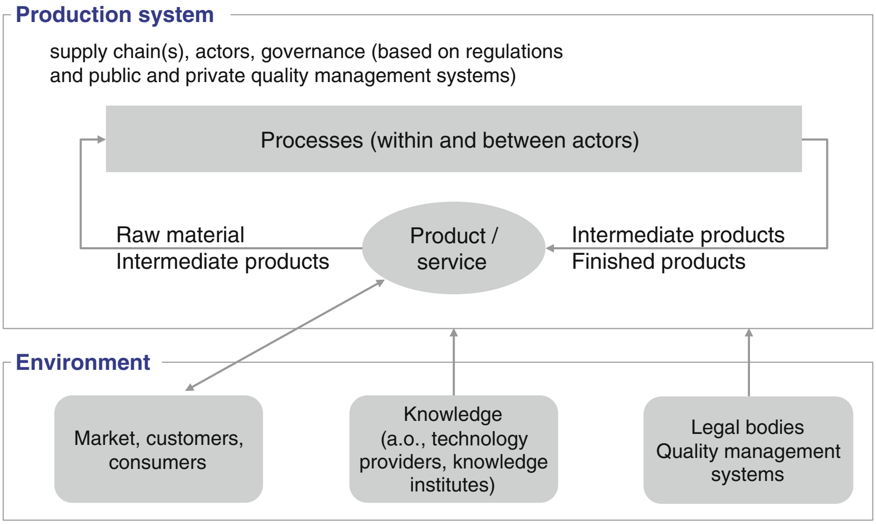


Fig. 2.4 Production system

In Sect. 2.4.2 properties are described of a food production system that is focused on production of quality food, which is often a differentiating characteristic to convince consumers to buy the food.

2.4 Concurrent Engineering in the Food Industry

The food industry, like any other industry, needs to innovate products and processes continuously to remain competitive. Although the food industry has been rather conservative in the past, in the present time the situation has dramatically changed. The food industry has become very complex and dynamic due to increasing product proliferation for serving the ever diversifying and globalising markets [34]. In additions, many incidents have occurred in the recent past, which has damaged consumer trust, requiring actions for producing safe and healthy products. Moreover, many different actors play a role in growing and producing food. Without collaboration between producers, sellers, legislators, etc., a new product is expected to fail. The food industry, like other industries, needs a CE approach for innovation in its products and processes. Before innovation in food is discussed in more detail, properties of the food industry are briefly discussed.

As indicated above, in the last decades several small and large incidents occurred. In the United States, for example, contaminated food causes up to 76 million illnesses, 325,000 hospitalisations, and 5,000 deaths each year [35]. Such and other major incidents that have taken place recently, like the BSE crisis in 1996 and Aviaire influenza in 2003 [36], have raised consumer awareness. More recently, incidents such as the EHEC crisis in 2011 and the mixing of horsemeat with bovine product in 2013 were reported. These incidents damage reputation of food supply chains and reduce consumer trust. Only with much effort consumer confidence can be restored (see e.g., [37]). Consumers increasingly wish to be convinced of the safety and healthiness of food products, which requires food supply chains to be transparent, while traceability of the materials and products needs to be guaranteed. In current global food supply chains this is a big challenge.

Many food supply chains (FSC) act globally. Consequently, worldwide involvement, regulations, requirements, and consequences of actions and decisions need to be taken into account. These differences necessitate additional safeguard for guaranteeing food safety, like specific quality management systems or tracking and tracing systems (see e.g., [38]).

Consumers increasingly require that the intrinsic attributes of the food products they buy are above threshold levels, such as safety, healthiness, colour, and taste. They also more and more focus on extrinsic attributes as well, demanding production processes to incorporate the three sustainable Ps [38]: *People*, availability of a good workplace for people and minding the welfare of animals on farms; *Planet*, environmental care by reducing pollution and the negative impact of pesticides and antibiotics; *Profit*, economic viability and profitability. Not only consumers demand increased sustainability, but institutions and organisations, like governments, environmental organisations, financial institutions, academic institutes, and supply chain actors more and more (are forced to) focus on sustainability.

Many different markets can be distinguished [39]. Each type of market poses specific requirements. Food supply chains need to differentiate themselves to satisfy these different demands. This means also, that structure and organisation of food supply chains need to be different. A supply chain for a stable, high-volume, product will be different from a low-volume, highly specialised, product (see e.g., [40]). Most stable, high-volume, products can be produced in supply chains with a market-type of governance structure (see also Sect. 2.4.2), provided that the minimum level of quality is guaranteed by means of quality management systems that apply to all or most actors in the supply chain [33]. The market influences the type of product that will be developed and, consequently, the organisation of the production system, the specific food supply chain that will produce, sell, or take back the product.

Food supply chains have specific properties that pose many constraints on managing the flow and quality of products. For example, products of food supply chains are perishable and may show considerable differences due to biologic variety even when genotypes and production processes are standardised. Food supply chains may consist of many, often small, actors, making collaboration and alignment more difficult. In addition, margins in the food sector are small, especially in conventional food supply chains with mass products, requiring supply chain actors to improve efficiency of their processes.

Food supply chains are also subject to rules, regulations, and quality management systems that exist to ensure food safety. Rules, regulations, and quality management systems can be found on European Union (EU), world, and national public level, but also on private level, issued, owned or monitored by associations, cooperatives, or individual companies. This situation is comparable to other high-risk areas, like medical products and aviation. Although basic quality has been guaranteed, at least in the EU, additional measures are often needed. For example, several supermarket supply chains demand specific quality from their suppliers with systems like GlobalGap (www.globalGAP.org). Especially for global food supply chains compliance with rules, regulations and quality management systems that exist in the different continents and countries involved is highly demanding.

Another complicating factor is that consumers demand product availability in broad assortments year-round at competitive prices. Changes in the area of trade laws (WTO) have led to more open markets. As a result, a large increase in cross-border flows of livestock and food products can be observed.

As already indicated above, to satisfy the different market and consumer demands, safeguarding or improving reputation and image without increasing costs too much, while remaining sustainable, FSCs need to continuously innovate their products, processes, and organisation. This will hardly be possible without collaboration between relevant stakeholders and taking into account specific properties of the food supply chain.

In Sect. 2.4.1 the CE system in the food industry with its different aspects and elements is discussed. Some examples of innovation activities are presented in terms of the framework presented in Sect. 2.3. In Sect. 2.4.2 essential properties are discussed of the food production system that is the result of or needs to be taken into account in a CE system in the food industry. Many food products are developed with a specific quality claim. To market and produce such products and to maintain the required quality level, safeguards are needed in terms of risk management through contracts. Two extremes of such contracts will be briefly discussed.

2.4.1 Innovation Processes in Food

The development of new products in food supply chains requires an open innovation approach (see Sect. 2.2.3). Involving relevant stakeholders is important for reducing the chance of failure of the new (or adapted) product. For example, changing the package of a meat product requires involvement of the package producer, the consumer, as well as the production of the content of the package. In addition, the packaging machine may also be affected requiring the technology provider to be involved.

Innovation processes must address not only the design of a new or adapted product, but also its market, production process and organisation. Referring to Fig. 2.3, an innovation process often starts with a new idea, which is often a product idea, but can also be an idea for a new market, a new process or new supply chain organisation. The idea may have originated from the creation of new technology or knowledge or may have been formulated by stakeholders like actors of an existing supply chain or consumers, but may also have been enforced by stakeholders like regulative bodies.

The innovation process needs to be well organised. This organisation is often different from the production system resulting from the innovation process. As discussed in Chap. 18, an innovation process requires the free exchange of knowledge to stimulate creativity. However, knowledge may be a critical asset for some companies involved, hampering the free exchange of knowledge. Suitable governance structures are needed to coordinate innovation processes and protect knowledge misuse and leakage when unwanted (see e.g., [41]).

The food production system will eventually produce the product envisioned in the innovation process. The production system, whether already existing or new, needs to be taken into account in a CE process and will gradually be redesigned or realised. Referring to Fig. 2.4, an innovation requires a reorientation on the markets, customers and consumers that will be served. For example, when a product is intended for a regional market only, the supply chain will be rather small and needs to be well aligned to guarantee the quality required. When the product will be made from several ingredients and raw material, the number of supplies can be large and may involve many different supply chains around the world. A new type of fruit or vegetable will need a customer for using or selling the product, while its supply needs to be guaranteed. Trading restrictions and quality demands may heavily impact a steady supply. When an existing product is adapted, the supply and production processes may need to be adapted too to satisfy new demands.

Innovation in food requires thorough understanding of the customers involved. Many new products fail [42], because consumers are not fully understood beforehand. Quality perceptions may change over time. Consumers must be able to perceive that products have a number of desirable properties. Before they are willing to buy new products, they must be able to infer these properties from appropriate cues. These cues are pieces of information, which the consumer uses to make an inference about quality. For example, colour and fat content of meat are an indicator of taste and tenderness. Packages may contain information on intrinsic and extrinsic properties of food products. Brands may give information on quality levels, origin, and production standards.

Innovation in the food area concerns many different product categories. First of all, food products are mostly fresh products, but other products can be preserved for a longer life, like canned or dried products. They can be unprocessed, like fruit and vegetables, but also processed, like sausages or pizzas, according to recipes and the addition of herbs, spices, and other ingredients. Suppliers of fresh products, the growers and farmers, may be involved in the development of new products, like new species resulting from biotechnology of genetics. Processing companies may be involved in new product development requiring the development of new recipes with possibly new ingredients and processing technologies. Often, panels for testing and tasting the new product are installed.

Like in other areas, tools and techniques can be used in the innovation process, such as brainstorming techniques and group decision tools. However, the use of technologies like QFD is limited in the food area. The complexity of food products, the many interactions between ingredients, and the influence of processes on functional properties of the product make it hard to fully apply the technology [43, 44]. The first matrix, however, the House of Quality, is useful to get insight into information necessary to make trade-off decisions and improve the product [43]. In addition, the matrices indicate links between quality characteristics as demanded by the consumer and actors in the production chain [43]. QFD needs to be adjusted to be applicable in the food industry, for example by allowing intervals for target values. In addition, most food ingredients are often physiologically active materials and, hence, still subject to change [43].

The development of new products often requires that new technologies need to be developed. These technologies may be needed on farms for monitoring and controlling growth and health. They may also be needed in processing firms like slaughterhouses, food processing firms, or distributors. Below, two examples are presented of the role technology can play in a food supply chain. The innovation processes are discussed that are needed to develop new technology and use it in a supply chain. The innovation processes are framed with the help of the framework discussed in Sect. 2.3.

Example 1: On the farm, technologies like RFID may affect effectiveness and efficiency. For example, sows producing piglets on a pig farm are nowadays often housed in groups. To manage feeding and monitoring pregnancy, RFID chips are used in ear labels together with automated feeding machines that regulate feed intake by sows, depending on the status of pregnancy. When giving birth is near, a sow is led into a separate area through a special gate of the feeding machine. Also, new growing techniques for fruits and vegetables will influence farming. For example, RFID sensors in vineyards will support farmers in regulating water supply (see e.g., [45]).

The first innovation process that can be distinguished is the development of suitable RFID chips. This innovation process needs to be specified:

1. The goal of development should be clear. Will the chip be used for process control in a pig stable or will it be used for climate control in a field outdoors? What impact will it endure from its environment? Will the chip be disposed of after use or will it be reused? Is additional effort needed to reduce its impact on the environment? What are constraints on size and weight? What are maximum costs for the chips to allow any profit? In defining the goal and constraints, not only the technology provider, but also representatives of farmers, need to be involved.
2. The process of focus in the example is the development of the chip by the technology provider. Trade-offs need to be made between a general chip and a dedicated chip. A chip applicable in more than one application domain, e.g., not only in pig farming, but also in monitoring cow performance, will be more expensive at first, but might have a larger gain in the longer term for the provider. Development of a generic chip requires the technology provider to

involve representatives of different application areas. In addition, material suppliers may play a large role as well.

3. In the example of an RFID chip for sow management, rules and regulations exist for limiting the number of physical interventions in an animal. This means, that the chip should be capable of replacing existing identification labels. In addition, the use of artificial material for the casing of the chips is not only impacted by regulation, especially environmental rules, but also by the environmental condition in which the chips will be used. The innovation process needs to take into account these regulations as well as the constraints put by the application area. In addition, the financial situation plays a role. What can the company invest itself? What amount must be borrowed? The innovation process of the example will consist of all stages, from idea development to prototype, when the provider does not produce a chip already. Otherwise, the innovation process will consist of adapting an existing product to make it suitable for one or more specific application areas.
4. The actors in the example will be the technology provider, representatives of farmers, and material providers in first instance. Knowledge is needed of material involved, RFID technology and its potential, and of farming conditions in the application areas.
5. The technology provider needs tools to design a chip and produce and test a prototype. It also needs tools and techniques for communicating with the other actors. Business cases may be used to investigate the feasibility of a particular design or pilot projects in an application area.

The output of the innovation process consists of the chip design, the service surrounding it (e.g., a data service for housing data read from the chips, or a recovery process for reusing obsolete chips) and the production system that markets, produces and maintains or takes back the particular chips.

The second innovation process that can be distinguished in example 1 is the process innovation that is needed in the application area that adopts the RFID chip. In the example of sow management, the farmer needs tools for applying the chips in the ear of the sow. He/she needs a handheld reader for reading the information on the chip: identification of the sow. The identification is connected to an information system with information, e.g., on the age of the sow, number of deliveries, insemination date, pregnancy duration, and necessary feed intake. The adoption of the RFID chip will dramatically change the sow management process. Group housing is possible for pregnant sows as well as dynamic feeding regimes. The innovation process can be specified with the 5-step approach:

1. The goal of the innovation process needs to be specified. In the example it could be increase of animal welfare, cost reduction in feed administration, time reduction in managing sows, increased efficiency in the whole process.
2. The process of focus is the change process from the old situation to the new situation. In this process the technology provider plays a large role including one or more information system providers, the farmer, and possibly the feed

provider. Accreditation bodies are also involved, because the resulting process must comply with existing regulation and quality management systems.

3. Sow management is subject to regulation and quality management systems, which specify the minimum housing conditions for individual and group housing. Sow management is the process of tracking progress of insemination and pregnancy, housing sows ready to deliver, housing and feeding sows with piglets, moving sows when piglets have been weaned, managing these sows to get them ready for new insemination. The process is an aspect of pig farming both for breeding of particular pig species and for production of pig meat. Pig farming for meat production consists of growing piglets and fattening pigs until they are ready for slaughter. The quality of piglets may increase by better sow management. The innovation stops when the change has been accomplished.
4. The actors involved in the change process are the farmer, the technology provider, an information system provider, and possibly the feed producer and stable builder. The farmer takes the lead, because it is his process that needs to be adapted.
5. Preparation of stables is needed, requiring the building of new housing or rebuilding existing ones according to the guidelines of regulation and quality management system. Pilot projects may be used to test the new process in a small group of sows. The information system may be gradually built or may be an existing system is installed and tuned to the situation. Additional modules may gradually be added.

The result of the innovation process is a changed production system, the production of piglets.

The two discussed innovation processes co-exist and interact, but they need to be viewed analytically as separate innovation processes, because, although actors may overlap, they play different roles in both innovation processes.

Example 2: In the processing stage of food production new technologies will affect the production process. An interesting example is the level of automation that can be achieved in the slaughtering of pigs. Horsens slaughterhouse of Danish Crown in Denmark has achieved an impressive level of automation in the first stage of the slaughter process where pigs are killed with CO_2, cleaned, and opened, intestines are removed and carcasses are cut in half before they are stored for 24 h in a cooling area (for a demo, see http://danishcrown.com/Danish-Crown/Welcome-to-the-virtual-slaughterhouse.aspx). The consequence of such an approach is that farmers need to adopt a higher level of standardisation with respect to the genotype of pigs they use, the feeding regime, and farm management. In this way weight and size of pigs as supplied to the slaughterhouse remain within more standard ranges as required for the machines in the slaughter line. Another example is a new cooking device, such as the Nutri-pulse e-Cooker, developed by IXL (www.innovation-xl.com), (see [46]), dramatically influencing the processing of food by increasing sustainability and safety in use. The cooker reduces cooking time at low temperature with minimal effect on proteins, vitamins, antioxidants, sugars, and other (healthy) substances.

As in example 1, the development of the automation technology is the innovation process of the technology provider together with the slaughterhouse. The technology may be an adaptation of existing technology or may have to be specifically developed for the customer at hand, the slaughterhouse. The production system eventually is the process of marketing, producing, selling, installing, and maintaining the automation equipment in the slaughterhouse.

The slaughterhouse itself undertakes a rather encompassing change process by adopting, installing, and using the automation equipment. A suitable implementation strategy has to be chosen to enhance the chance of success. Because margins in pork production are rather small, any disturbance of daily operation may have large consequences. The change must, therefore, be planned and executed carefully. Moreover, since investments are huge, continuity of the business needs to be ensured as much as possible for the longer term. Actors involved are various, from process managers of the slaughterhouse, to technology providers (automation equipment for slaughtering, slaughter-line experts, etc.), farmers, feed companies, legislators, customers, like food processing companies, etc. The result of the innovation process in the slaughterhouse is an efficient slaughter process with limited number of people in the line and reduced floor space.

Another promising area for innovation in the food area is the development of chain-wide information systems. The need for providing consumers with safeguards on the safety of food and enabling the fast recovery of errors in the food supply chain, the exchange of information in the supply chain is essential. Chain-wide information exchange reduces the asymmetry of information between actors in a supply chain. Although there are successful examples of chain-wide information systems, many barriers also still exist. Successful examples are often systems that involve only part of a supply chain, like health management systems on a farm relating farmers, slaughterhouse, veterinarian, and possibly the feed company (see e.g., wikiporc: www.wikiporc.fr).

Chain-wide information systems for traceability are more difficult to realize. In a recent project the feasibility of a chain-wide information system is investigated based on the application of RFID on pig farms and DNA profiling for tracing the origin of a piece of pig meat (see [47]). It appeared that the supply chain studied is not yet ready for adoption of such a system, while some farmers were already convinced of the use of RFID for enhancing effectiveness and efficiency with RFID on their farms. Local optimization by improving farm processes might lead to profit for farmers, but the gain might be larger when the impact on the whole downstream supply chain is also recognized and measures are taken to optimally benefit from the efficiency and quality gain on farms. DNA profiling was seen as profitable, but requires investments in a unique population of boars used for insemination. As already indicated above, margins in the food area are regularly small. Investments in chain-wide changes require, moreover, the involvement of many supply chain actors, which may each have a different interest in participating in the innovation. Costs and benefits may also be different in different stages of the supply chain. Only when these costs and benefits are equally shared, innovations are more likely to happen.

Many new food products are developed for a specific quality label (see e.g., the farmer cooperative BESH, www.besh.de). Supply chains producing products with a specific quality label need to be organized well to maintain the quality of the product as claimed in the label and to prevent image loss. In the next section, examples are given of supply chains that have been organized for production of quality products. The examples are taken from case studies in the domain of pork supply chains in Europe.

2.4.2 A Food Production System—Integration and Coordination

As explained above, the result of an innovation system is a production system. Innovation in food mostly leads to a production system in the form of a food supply chain. It is important for an innovation system to understand the properties and constraints of a food supply chain to develop a production system that is sustainable in terms of profit (the product must keep value for the consumer), people (food must be safe as well as the process conditions for people and animals), and planet (impact on the environment must be as small as possible). Food supply chains are subject to many demands and constraints like any other high-risk area like the medical area and aerospace. In this section, integration and coordination mechanisms of food supply chains are discussed that are needed for achieving and maintaining the quality level required for keeping consumer interest in buying and enjoying the food products. Examples are give of different supply chain structures.

Worldwide, food products have to comply with legislation and quality demands. In developing new products and the processes for producing them, legal and quality rules play a large role. Many products are developed with a specific quality claim. For example, three EU quality systems exist to promote and protect food products (http://ec.europa.eu/agriculture/quality/schemes/index_en.htm). These are protected designation of origin (PDO—covers agricultural products and foodstuffs which are produced, processed and prepared in a given geographical area using recognised know-how), Protected Geographical Indication (PGI—covers agricultural products and foodstuffs closely linked to the geographical area. At least one of the stages of production, processing or preparation takes place in the area), and Traditional Specialty Guaranteed (TSG—highlights traditional character, either in the composition or means of production).

Legislation and quality demands, specifically the public ones on European as well as national level, set a minimum level of quality that should be achieved as a reaction to the already mentioned food crises. For example, the European Union has issued the General Food Law (GFL), which emphasises that firms hold primary responsibility for quality in the chain. Many chains go beyond this basic level to distinguish themselves to the end consumer by setting additional, often private, quality demands. Maintaining the additional quality level needs additional efforts to avoid reputation disasters. The question that can be raised is what integration and coordination mechanisms are needed in different chains with different quality and market characteristics. Integration and coordination between supply chain actors is needed to cross boundaries between the different functions, processes, and material and information flows in a supply chain [48].

Integration can be expressed in terms of inter-company relationship structures. A supply chain with a minimum level of integration shows mainly market relationships, while a high level of integration exist in vertically integrated supply chains with tightly controlled relationships. In between these two extremes various hybrid forms exist as depicted in Fig. 2.5. At the left side of the continuum companies in a supply chain are relatively independent, while at the right side there is strict control over the flow of goods and information. From left to right various forms of integration can be found from informal (long-term) relationships to formal written contracts and long-term collaboration agreements.

To achieve integration coordination mechanisms are needed to manage the flow of information and materials and to take decisions. Examples of coordination mechanisms are standardisation of output, process, or knowledge and skills. More expensive and complex coordination mechanisms to achieve integration in a supply chain are coordination by hierarchy or plan and coordination by the creation of lateral linkages [49, 50]. With respect to relationship structures as depicted in Fig. 2.5, coordination tends to be more complex and expensive going from left to right.

The quality level that supply chain actors together want to achieve may influence the degree of supply chain integration required. Quality management systems in general provide the standards and monitoring mechanisms for achieving, maintaining, or improving the desired quality level and to communicate quality across the supply chain and to end consumers. Quality standards need to comply with, but often extend EU, national, and sector rules and legislation. We may distinguish public from private quality management systems based on ownership of the quality standards. In addition, quality management systems may apply to a whole supply chain or to single supplier-client, or company-to-company, relationships. Monitoring of compliance to the standard is performed by either the owner of the standard or by an external auditing agent. Ownership determines the decision authority and flow of necessary information. Information is also needed to assess compliance to the standard. Finally, only a few actors in the sector or a large part of supply chain actors may adopt a quality management system.

Below, we describe some examples of pork supply chains in the EU based on case-study research performed in the EU (EU-FP6-036245-2) project Q-Porkchains. The examples show different integration and coordination mechanisms related to the quality management system(s) adopted by the respective supply chains. More examples can be found elsewhere [51].

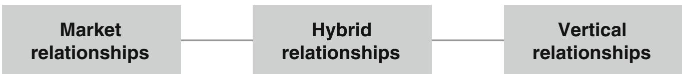


Fig. 2.5 Range of supply chain relationships

Example 1: Private chain-wide quality management system as industry standard.

Supply chains with this type of system have a private chain-wide quality management system on top of the baseline quality standards set and monitored by the EU, the state, or other public actors. Most actors in the whole industry sector have adopted the private chain-wide quality management system. In this sense, the system can be considered as the industry standard. In addition to the chain-wide quality management system, chain actors my set private standards for the immediate linkages in the chain on top of the chain-wide quality management system. These additional link-to-link standards have also been widely adopted in general by the respective horizontal stages of the supply chain. This type of supply chains can be found in the fresh pork meat industry in Germany, where QS (Qualität und Sicherheit) is the chain-wide quality management system, the fresh pork meat chain in The Netherlands, where Integrated chain control (IKB) is the chain-wide quality management system, and in the fresh pork meat chain in France, where VPF (Viande de Porc Français) is the chain-wide quality management system.

In the Dutch fresh pork meat chain contracts are relatively rare, although different relationship structures can be found at different stages of the supply chain. Contractual relationships mainly exist in the breeding stage, while free trade is found in farmer-slaughterhouse relationships. Most relationships can be characterised, though, as informal and long-term. The Dutch government sets baseline standards for the sector in accordance with EU legislation, but even exceeds EU legislation, for example with respect to animal welfare. Additional standards have been set by the private society of pig companies 'De Groene Belangenbehartiger' and by the Product Board of Cattle and Meat (PVV). In both groups all supply chain stages are represented. The standard is called IKB (Integraal KetenBeheer—Integrated Chain Control). IKB is widely used in the Dutch pork industry: more than 90 % of the pigs produced in The Netherlands are IKB pigs. IKB is a chain-wide quality management system. It sets requirements for each linkage in the chain. Chain actors may put additional demands on top of IKB requirements.

Compliance with the IKB standard is outsourced to a third party certifying agency, like Lloyds and SGS. In addition, large chain actors, like retailers, undertake their own inspection of their direct suppliers. IKB is only communicated in inter-chain linkages. Retailers use their house labels to communicate quality to end consumers. With respect to coordination, IKB acts as standardisation of processes and outputs. Since the supply chain is very large with many actors and supply relationships, it is unfeasible for one actor to coordinate the whole supply chain.

New products in this type of meat supply chain need to be suitable for a mass market. New products can be fresh, like a new meat cut, or meat sold in a package with ingredients that can be used for cooking. Products can also be processed. Examples of processed products are hamburgers, shoarma, or sausages.

Example 2: Public chain-wide quality management system.

Supply chains with this type of system have adopted a (voluntary) public chain-wide quality management system on top of the baseline quality standards set and monitored by the EU, the state, or other public actors. These public chain-wide quality management systems are mostly regional systems, like Protected Designation of Origin (PDO) and Protected Geographical Indicator (PGI). Such systems tie production to a specific region. Within these regions, the quality management systems may be widely used. The Spanish Iberian PDO supply chain is an example of a supply chain in this category.

All chain actors have signed contracts with the control board PDO Guijuelo, which coordinates the supply chain. Between chain actors, market relationships exist with long-term relationships. The control board Guijuelo is an independent regulatory council responsible for setting the PDO standard and monitoring compliance with this standard. PDO standards are protected by EU legislation. This protection is assigned only when certain strict conditions are met. Most importantly, the product characteristics must be (partially) determined by or linked to the specific geographical location [52]. The PDO must satisfy in addition European and Spanish regulation for meat production. The PDO Guijuelo quality standard is owned by the regional government, which also monitors compliance with the standard next to independent monitoring agencies. The PDO label is communicated in intra-chain relationships as well as to the end consumer.

With respect to coordination, the control board PDO Guijuelo uses coordination by hierarchy and plan. Standardisation of processes and outputs is provided by the PDO standard, making market relationships possible.

The PDO standard applies to the Iberian ham sold worldwide. Innovation in this type of product is limited, because of the recognised taste and quality of the product. However, new products may be developed based on Iberian pork, fresh as well as processed. The Iberian pig contains more than only ham. Other parts of the pig are sold as fresh meat to restaurants and special butchers.

The examples that have been briefly described above show that different quality management systems exist with different chain governance structures and coordination mechanisms. In developing supply chains with suitable structures not only the bi-lateral relationships between actors need to be taken into account, but also the upstream and downstream relationships to prevent suboptimal arrangements [53].

2.5 Summary and Further Research

At the time CE emerged, the importance of cross-functional thinking was gradually recognized. Since then, in particular medium and large companies have adopted CE practices [54, 55]. Although many terms have been used for cross-functional and cross-border collaboration, the essence has been the formation of teams in which also the customer and supplier are involved [56, 57]. Information technology plays a key role in facilitating information sharing between the many actors involved [58].

Although the term CE has gradually disappeared, it has set the scene for further development of collaborative product development in networks and supply chains of essential stakeholders [59]. By adopting CE the necessary skills/competencies have become in place for gradually moving towards CE*, ESI, and CI to manage the complexity of continuously innovating products and processes [57].

New theories or frameworks need to be developed to identify patterns within or between industry sectors and product development project types. Several attempts have been made in the past, while new challenges have also emerged. In this chapter CE has been described as an innovation system aimed at generating a production system that is capable to sustainably produce the new product envisioned. The production system is often also a complex system of collaborating companies, for example in a supply chain. Based on this view a framework has been proposed that can be used to describe CE processes and its resulting production system for further analysis. Examples of innovation in the food industry have been presented, which show that several innovation systems may co-exist and interact. In addition, specific properties of food production systems, in particular integration and coordination, have been presented that need to be taken into account when developing a production system.

Since the framework for framing the system of CE and the accompanying production system presented in this chapter is rather abstract and generic, it needs to be made specific for different application areas. For example, planning and scheduling techniques may help to more specifically identify the order of activities and their timing. Performance management methods may help to determine useful indicators for managing performance of the CE process. In Chap. 8 of this book, an example of a more specific framework is presented.

Specific attention is needed for the properties of the resulting production system. The examples from the food area have shown that specific requirements exist for food production systems. These may be different for production systems in other application areas, given the nature of supply chains and necessary arrangements, like contracts, referring to the necessary degree of integration and coordination, in these areas. Theories of coordination, social-dynamics, organisation structures, etc., may help to further develop the description of the production system.

References

1. Trygg L (1993) Concurrent Engineering practices in selected Swedish companies: a movement or an activity of the few? J Prod Inn Manag 10(5):403–415
2. Clark KB, Fujimoto T (1991) Product development performance: strategy, organization, and management in the world auto industry. Harvard Business School Press, Boston
3. Winner RJ, Pennell JP, Bertrand HE, Slusarczuk MM (1988) The role of concurrent engineering in weapons system acquisition. IDA R-338. Institute for Defense Analyses, USA
4. Bergstrom RP (1990) Assembly, promises and simultaneous engineering. Prod 102(2):50–56
5. Cleetus J (1992) Definition of concurrent engineering. CERC Technical Report CERC-TR-RN-92-003, Concurrent Engineering Research Center, West Virginia University, USA
6. Koufteros X, Vonderembse M, Doll W (2000) Concurrent engineering and its consequences. J Op Manag 19:97–115
7. O'Neal Ch (1993) Concurrent engineering with early supplier involvement: A cross-functional challenge. Int. J Purch Mat Manag 29(1): 2–9

8. Eden C, Williams T, Ackermann F, Howick S (2000) The role of feedback dynamics in disruption and delay on the nature of disruption and delay (D&D) in major projects. J Op Res Soc 51:291–300
9. Terwiesch C, Loch CH, de Meyer A (2002) Exchanging preliminary information in concurrent engineering: alternative coordination strategies. Org Sci 13(4):402–419
10. Chen Y-M, Liang M-W (2010) Design and implementation of a collaborative engineering information system for allied concurrent engineering. Int J Comp Int Manuf 13(1):11–30
11. Lu SC-Y, Elmaraghy W, Schuh G, Wilhelm R (2007) A scientific foundation of collaborative engineering. Ann CIRP 56(2):605–633. doi:10.1016/j.cirp.2007.10.010
12. VIVACE (2005) Collaborative methods and tools. VIVACE Forum, CERFACS, EADS, Warwick
13. Lenox M, King A, Ehrenfeld J (2000) An assessment of design-for-environment practices in leading US electronic firms. Interfaces 30(3):83–94
14. Todd S (1992) Collective action: theory and application. University of Michigan Press, USA
15. Willaert SSA, de Graaf R, Minderhoud S (1998) Collaborative engineering: concurrent engineering in a wider context, A literature review. J Eng Techn Manag 15:87–109
16. McIvor R, Humphreys P (2004) Early supplier involvement in the design process: lessons from the electronics industry. Omega 32:179–199
17. Zsidisin GA, Smith ME (2005) Managing supply risk with early supplier involvement: A case study and research propositions. J SCM Fall 41(4):44–57
18. Chesbrough H (2004) Managing open innovation. R Techn Manag 47(1):23–26 Jan–Feb
19. Dodgson M, Gann D, Salter A (2006) The role of technology in the shift towards open innovation: the case of Procter & Gamble. R&D Manag 36(3):333–346
20. Sarkar S, Costa AIA (2008) Dynamics of open innovation in the food industry. Trends Food Sci Technol 19:574–580
21. Vanhaverbeke W (2006) The inter-organizational context of open innovation. In: Chesbrough H, Vanhaverbeke W, West J (eds) Open innovation: researching a new paradigm. Oxford University Press, UK
22. Ainscough M, Yasdani B (2000) Concurrent engineering with British industry. Concurrent Eng 8(1):2–11
23. Boer H, Krabbendam JJ (1993) Inleiding organisatiekunde (Introduction to organisational science). University of Twente, Enschede, The Netherlands
24. Wognum PM, Faber ECC (2002) Infrastructures for collaboration in virtual organisations. Int J Netw Virt Org 1(1):1–23
25. Schein H (1996) Culture: the missing concept in organisation studies. Adm Sci Quart 41 (2):229–240
26. Scott WR (1995) Institutions and organisations. Sage, Thousand Oaks, p 178
27. Giddens A (1984) The constitution of society: an outline of the theory of structuration. University of California Press, Berkeley and Los Angeles
28. Flood RL, Jackson MC (1991) Creative problem solving: total systems intervention. Wiley, Chichester, p 250
29. Boer H (1990) Organising for manufacturing innovation: the case of flexible manufacturing systems. PhD Thesis, University of Twente, Enschede, The Netherlands
30. Weerd-Nederhof PC de (1998) New product development systems, operational effectiveness and strategic flexibility. PhD Thesis, University of Twente, Enschede, The Netherlands
31. Faber ECC (2001) Managing collaborative new product development. PhD Thesis, University of Twente, Enschede, The Netherlands
32. Kerzner HR (2013) Project management: a systems approach to planning, scheduling, and controlling. Wiley, USA
33. Wever M, Wognum PM, Trienekens JH, Omta SWF (2010) Alignment between chain quality management and chain governance in EU pork supply chains: a transaction-cost-economics perspective. Meat Sci 84:228–237
34. Trienekens JH, Wognum PM, Beulens AJM, van der Vorst JGAJ (2012) Transparency in complex dynamic food supply chains. Adv Eng Inf 26(1):55–65

35. Smith-DeWaal C (2003) Safe food from a consumer perspective. Food Control 14:75–79
36. Plaggenhoef W van (2007) Integration and self regulation of quality management in Dutch agri-food supply chains. PhD Thesis, Wageningen University, Wageningen, The Netherlands
37. Greenberg J, Elliot Ch (2009) A cold cut crisis: listeriosis, maple leaf foods, and the policy of apology. Can J Comm 34:189–204
38. Wognum PM, Bremmers H, Trienekens JH, van der Vorst JGAJ, Bloemhof JM (2011) Systems for sustainability and transparency of food supply chains—current status and challenges. Adv Eng Inf 25(1):65–76
39. Grunert KG, Wognum N, Trienekens J, Wever M, Veflen Olson N, Scholderer J (2011) Consumer demand and quality assurance: segmentation basis and implications for chain governance in the pork sector. J Chain Netw Sci 11(2):889–897
40. Franks J (2000) Supply chain innovation. Work Study 49(4):152–155
41. Gambade PJP (2014) Management of innovation in networks and alliances. PhD thesis, Wageningen University, Wageningen, The Netherlands
42. Grunert KG, Valli C (2001) Designer-made meat and dairy products: consumer-led product development. Livest Prod Sci 72:83–98
43. Benner M, Linnemann AR, Jongen WMF, Folstar P (2003) Quality function deployment (QFD)—can it be used to develop food products? Food Qual Pref 14:327–339
44. Costa AIA, Dekker M, Jongen WMF (2001) Quality function deployment in the food industry: a review. Trends Food Sci Technol 11:306–314
45. Matese A, Di Gennaro SF, Zaldei A, Genesio L, Vaccari FP (2009) A wireless sensor network for precision viticulture: the NAV system. Comput Electron Agric 69:51–58
46. Foodvalley NL (2013) 50 selected food innovations. Foodvalley NL, Wageningen
47. Wognum PM, van Erp T (2013) TIVO—Traceability of individual pigs in the organic chain. Report GO-EFRO project TIVO. Wageningen University, Wageningen, The Netherlands
48. Romano P (2003) Coordination and integration mechanisms to manage logistics processes across supply networks. J Purch Suppl Manag 9(3):119–134
49. Mintzberg H (1979) The structuring of organization—a synthesis of the research. Prentice-Hall Inc, Englewood Cliffs
50. Thompson JD (1967) Technology and structure. Organizations in action. McGrw-Hill, New York, pp 51–65
51. Trienekens J, Petersen B, Wognum N, Brinkmann D (eds) (2007) European pork chains. Diversity and quality challenges in consumer-oriented production and distribution. Wageningen Academic Publishers, Wageningen
52. Raynaud E, Sauvee L, Valceschini E (2005) Alignment between quality enforcement devices and governance structures in the agro-food vertical chains. J Manag Gov 9:47–77
53. Wever M, Wognum PM, Trienekens JH, Omta SWF (2013) Supply chain-wide consequences of transaction risks and their contractual solution: towards an extended transaction cost economics framework. J Supply Chain Netw 48(1):73–91
54. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Sys Manag 6(4):307–323
55. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Sys Manag 7(2):101–115
56. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Sys Manag 7(2):116–131
57. Chang D, Chen CH (2014) Understanding the influence of customers on product innovation, Int J Agile Sys Manag 7(3–4):348–364
58. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Sys Manag 6(1):7–24
59. Peruzzini M, Germani M (2014) Design for sustainability of product-service systems, Int J Agile Sys Manag 7(3–4):206–219

Chapter 3
Complex Engineering Programs as Sociotechnical Systems

Bryan R. Moser and Ralph T. Wood

Abstract By framing complex engineering as sociotechnical systems, the concurrent engineering (CE) community can gain new insights, practices, and tools to cope with program difficulties. Todays distributed product development teams need to manage both human (organization) and technical (product and process) elements of their work. These sociotechnical elements combine in a real-world engineering program as an integrated architecture with dynamic interactions. Based on traditional representation and analysis of engineering activity, the prediction of performance can become challenging. Practices for engineering planning and ongoing management often rest upon deeply held beliefs of stability, detailed decomposability, and feasible control of related products, processes, and organization. However, while these assumptions drove collocated manufacturing during the industrial revolution, today's engineering programs—and how the CE community considers them—have evolved. This chapter provides historical context on the evolution of systems thinking as applied to engineering and project management. Concepts are summarized as forces which reinforce and those which restrain the treatment of engineering programs as sociotechnical systems. Complexities of real world engineering programs can be considered in order to anticipate emergent outcomes driven by dynamic interaction of technical and social characteristics. This perspective is leading to a new generation of methods and practices for high performance engineering programs.

Keywords Sociotechnical systems · Project design · Collaborative engineering · Simulation-based planning · Scheduling · Complexity · Teamwork · Learning

B.R. Moser (✉)
Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: bry@mit.edu

B.R. Moser · R.T. Wood
Global Project Design, 75 Arlington Street Suite 500, Boston, MA, USA
e-mail: ralph.wood@gpdesign.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_3

3.1 Introduction

The situation faced by engineering initiatives includes dynamic changes in our products, our teams, and how we work together. Judgment and embedded practices —built on decades of traditional work processes and system architectures—lose relevance as unanticipated project dynamics emerge, driven by new combinations of architecture and behaviors. In contrast, if systems, careers, and markets change within an organization's capacity to be aware and respond, then the embedded know-how and standards within an engineering practice may be maintained. Corresponding processes and system standards evolve so that a useful alignment amongst product, process, and organization remains. However, todays engineering work systems of systems do not evolve in such a way, nor does their complexity permit the sustainable adaptation assumed by traditional engineering project methods. Of course, if variation can be reduced, standards reinforced and reused, and careers maintained, the benefit of leveraged know-how is recognized.

Our research and industrial experience has been driven by root cause analysis of performance in dynamic work situations to significantly improve schedule, cost, and scope delivery. Sometimes a problem in an engineering project is simply a mistake which could have been anticipated and prevented based on traditional engineering and management methods. However, recent dynamics of complex projects yields surprises and variations undetected even by teams considered best performers. Rather than elimination of these unexpected variations by forcing the engineering project to fit anticipated form and processes, our research examines the systemic conditions which lead to these surprises and how the teams either ignore or become aware of the systemic consequences in a timely fashion.

Future engineering systems will continue to increase in complexity technically and organizationally. Arguably the only certainties associated with programs for developing, improving, or remediating major products systems are negative outcomes: the programs will likely not accomplish all of their objectives and/or they will overreach their desired schedules and costs. Our current world is replete with examples of first-time program failures, for instance, Boeing's 787 Dreamliner aircraft and Blackberry's Z10 and Q10 smartphones, not to mention programs for economic recovery, drug control and peace in various nations. Why is this so? Was the program plan at fault, or was there a problem with the execution of the plan? Was the program plan distilled from a variety of options, or was the plan formulated narrowly around a favorite concept? Did the plan rely on proven technology or were inventions that pushed the limits of nature involved? Under the pressure of cost and schedule, did the execution teams omit an essential validation step or did the teams not give adequate attention to, or lose sight of, a critical program area that carried multiple dependencies? Was the program plan designed?

Discovering antidotes to the issues raised by these and further questions motivates this chapter and our desire to assist the international concurrent engineering (CE) community advance the practice of CE for managing complex programs and projects. Our observation, elements of which we have demonstrated successfully for over 15 years, is that a program and the customarily distributed development teams associated with it form a sociotechnical system. After talking with one of the authors (BRM), Dr. Marietta Baba, Michigan State University, defined the problem we face in this way:

> A globally distributed team is a socio-technical system—combining both human and technical elements in dynamic interaction. The interplay [interdependency] of these forces is non-linear, meaning that it is extremely challenging to forecast team performance outcomes in today's turbulent business environments.

The complexity of a program compounds with the realization that the product or objective of the program is, or is part of, a larger sociotechnical system in which the technical aspects of the system cannot be divorced from the social aspects of the environment in which the system will be designed, developed, and deployed.

By applying systems thinking principles for sociotechnical systems to program management, we can cope with complexity and secure desired program results consistently. The simple name of our approach, Project Design, belies its deep underpinnings although contains a clue: program designing is different from traditional program planning. Project Design embodies three distinguishing features:

- A comprehensive design **representation**, with the aid of a graphical engine, of the architectural elements of a program and their interdependencies
- **Analytics** enabling teams to bound rapidly the future consequences and feasibility of program architecture design options—technical and social—and decisions
- **Social process** that fosters program team learning and inculcation of the cultural values, artifacts and beliefs needed to succeed.

This chapter opens with a context-setting section that traces the evolutions of program management and systems engineering. From there we explore the opportunities and benefits of designing complex programs as sociotechnical systems, particularly the implications for CE. A subsequent chapter (see Chap. 8) provides further details about the key program design and execution concepts of Project Design and compares this methodology with other contemporary approaches to program management.

3.2 What Makes Concurrent Engineering Difficult?

We characterize CE as "teamwork under complexity." Indeed, in contrast to CE, a traditional, sequential, fixed process approach to engineering requires simplicity; e.g. tasks are of fixed duration, dependence can be captured as precedence, and the behaviors of organizations are uniform and without error. However, modern work and product systems are inherently complex. Imposition of an engineering process which ignores these inherent characteristics can lead to unrealistic targets and poor results.

Teamwork involves cooperation among humans, which is made difficult by their distributed locations and time zones, by their diverse social and cultural backgrounds and languages, and by how humans are cognitively "wired" [1]. Complexity is borne of multiple dependencies between activities to be performed in the conceptualization, design and realization of an objective: a product, process, system, service or even a whole society [2]. These activity dependencies have their origins in the technical architecture of the objective and in the architecture of how program activities are allotted to teams and then performed. Generally, the more elements that the objective has, the greater will be the number of dependencies and the chance that several dependencies will be interdependent or concurrent, meaning that the progress of one activity depends on the progress of the other and vice versa. Indeed, the trend of human endeavors to encompass "systems of systems" fosters multiple interdependencies.

When interdependencies occur in a program, the course of execution can become non-linear and iterative, with no guarantee that the iterations will converge to the requirements of the objective. Another consequence, called detailed or generative complexity [3], can be that the program takes on emergent behaviors, which deviate from plan and confound the teams; again, the greater the number of components and changes in how they are related, the less the likelihood that outcomes will be predictable.

But all is not lost in the quest of improving CE. We conclude, from this brief discussion, that the iterative design of a program's architecture, including the technical architecture of the objective, together with the "social" architecture of the performing teams' task roles, attention, and interfaces, is a key to program success. If we could identify principles and some useful tools of architectural design in the context of a program or project, then planning and ongoing performance in engineering projects would appear to be a straightforward design activity. Another key to program success appears to be managing the sociocultural behaviors of distributed teams so that they generate and make the right choices for the many options that attend the objective. Designers (in this case program managers and development teams themselves) need to learn how to use what they already know [4], learn how to realize what they don't know, and learn how to learn what they need to know. Program teams are challenged to self-organize, to become competent in systems thinking and to practice "execution as learning" [5]. As such the elements of this perspective are "information bonded."

In the next three sections, we reinforce these notions by tracing their origins, evolution through organizational metaphors (the backdrop for program management), and systems thinking.

3.3 A Century of Work as Centrally Controlled Tasks

The roots of engineering project management began more than a century ago, with thought leadership by Taylor, Fayol, Gantt, and other industrialists seeking to define and improve the organization of work. The First Conference on Scientific

Management was held in 1911 at Dartmouth College. The essence of "scientific management", clearly framed by Taylor, was summarized by one of the key participants in the conference:

> The theory of the proper execution of work is that it should be **planned completely before a single move** is made, that a route-sheet which will show the names and order of all the operations which are to be performed should be made out and that instruction cards should be clearly written for each operation….
> By this means the order and assignment of all work, or routing as it is called, should be **conducted by the central** planning or routing department. This brings the control of all operations in the plant, the progress and order of the work, back to the central point.
> Kendall [6] (emphasis added)

The Gantt chart is one of the oldest and most widely used models of project work. Project managers today are often surprised to learn that the Gantt chart has been in use for more than 100 years. The Gantt chart embodies the view of work promoted by the pioneers of scientific management. Clark argues that the Gantt chart reinforces the need for a plan, consistent with Taylor's scientific management:

> The use of a Gantt chart makes it necessary to have a plan. Recording that plan on a chart where it can be seen by others has a tendency to make it definite and accurate and to promote the assignment of clear-cut tasks to individuals. [7]

Tasks are related through sequence. Workers are viewed as interchangeable resources. According to Clark, "the charts are so simple that they can be understood by anyone—even by a foreigner who cannot read the language in which they are written."

The Gantt chart has been extended over the years, including different views of a project in one chart. The most suggested use is to track a plan against actual work progress. Each task row might show two bars, one for the plan and one for actual work. Since the length of each task bar corresponds to duration in calendar time rather than (necessarily) progress in the work, an additional attribute measuring the progress from 0 to 100 % is sometimes included. While the Gantt chart has been extended, deeply held assumptions remain—of fixed task duration, pre-determined task sequence, and that work is to be planned in detail by experts before the project begins.

3.4 Evolution of Engineering Programs as Organization

Until the latter part of the twentieth century, we also find the evolution of program management within the history of organization development. Gharajedaghi identified three types of organizational constructs that trace the latter [8]: mindless system (mechanical model), uni-minded system (biological model), multi-minded system (sociocultural model). He introduced two modes of inquiry for each of these models to create six sub-models of organization. His modes of inquiry referred to the analytical approach, focused on independent variables, and on the systems approach, focused on interdependent variables.

Earlier, Morgan [9] advanced eight metaphors or images to enrich the understanding of organizational development:

- as mechanical systems
- as organisms
- as brains
- as cultures
- as political systems
- as psychic prisons
- as flux and transformation
- as instruments of domination

Although the content of the two authors' models doesn't always agree, taken together their treatises provide a comprehensive history. The point of the models is to follow how organizational thinking has evolved to current practice and ideas. As in many evolutions vestiges of earlier thinking have endured and can be found pervading today's organizational structures; this is particularly true of the mechanical model, which featured the organization chart and central command and control.

For the present purpose we trace the roots of program management to the mechanical model of organization defined by Max Weber, Frederick Taylor, Henri Fayol and others in the late nineteenth and early twentieth centuries. The blueprint for the structure is the organization chart. Work is routinized and controlled, like clockwork, by a central authority [10]. In program management the Gantt chart for scheduling, discussed above, is an artifact of the mechanical model. People are hired to operate the machine, and workers follow tightly prescribed activities. In this concept of organization, management provides thinking, planning and top-down control; workers "work." The majority of the organization is "mindless."

Fayol in France—working in parallel to Taylor in the United States—published *General and Industrial Management* in 1916 which outlines a comprehensive and centralized approach to management. He promoted five key roles for management: planning and forecasting, organizing, commanding, coordinating, and controlling. Consistent with the advocates of Scientific Management, Fayol argued against character and emotional driven leadership, instead laying out a system, with 14 principles, to be followed [11, 12]. Weber in contrast came from a philosophical rather than industrial viewpoint, through sociology arguing against ad hoc organization driven by character and instead exploring structure and bureaucracy as a basis for permanent and efficient organizations. He emphasized hierarchy and authority with a view towards very stable and clearly defined roles and process. His writings reflect a drive towards efficiency and are in large part consistent with others at that time considering the organization as a machine-like system [13]. Interestingly, Weber also observed the risk of organizations leading to a de-personalization of human individuals as they became a part of the larger whole.

Although the thought leaders of the mechanical model recognized the need for employee satisfaction and esprit de corps, these requirements were overshadowed by a quest for efficiency. Apart from workers themselves being treated like machines, the mechanical model of organization is grounded in the assumption of stability of both the external environment and the workforce; it is not adaptive to a changing business environment, and concepts of coordination, uncertainty and workforce mobility are absent. Rote adherence to routine displaced thinking and learning, two competencies that are critically important to today's organization; thus, the mechanical model of organization is not designed for innovation. Profit is its end goal.

In contrast, thinking about an organization as a living organism introduces the insight that an organization is an open system, which must interact with and adapt to its environment in its quest to survive. By contrast the mechanical model of organization is a closed system that operates independently from its external environment. One consequence of the open system property is that different styles (or "species") of organization are required to fit different environments; there is no "one-size-fits-all" prescription [14]. A viable organization in this model adopts growth as its performance measure of success, and the organization focuses on the adequacy of its resources to promote growth. Unlike the mechanical model of organization, profit is not an end goal but the means to achieve growth in the organic organization. Proponents of these ideas, advanced in the 1950s and 1960s, include Burns and Stalker, Lawrence and Lorsch, Boulding and Simon.

Herbert Simon wrote

> Organizations are not highly centralized structures in which all the important decisions are made at the center. Organizations operating in that centralized way would exceed the limits of human procedural rationality and lose many of the advantages attainable from the use of hierarchical authority. Real-world organizations behave quite differently [15].

More recently Daft, Robbins, and Burton continued to characterize the nature of organizations consistent with common notions of natural systems prevalent in the late twentieth century. Burton summarizes these views including definitions Robbins' definition:

> An organization is a consciously coordinated social entity, with a relatively identifiable boundary, which functions on a relatively continuous basis to achieve a common goal or set of goals (Robbins in [16]).

Burton goes on to describe an organization with five key characteristics:

- A social entity, a group of people
- A purpose and shared goals
- Activities
- Boundary, outside is the environment
- An organization is explicitly and deliberately constructed; "designed"

Burton argues that the design of the organization should be guided by the fit of the environment and the structure and properties of the organization. Behavioral properties of an organization include differentiation, formalization, complexity, centralization, span of control, rules, procedures, professionalization, meetings, reports, communications, media richness, and incentives [16].

At the system level the organization as organism is able to make choices, although its components cannot. The system is under the control of a single brain or executive function. By means of a communication network, the executive function receives information from sensing elements and then issues instructions to activate relevant components of the system. The components are programmed with feedback mechanisms that react in predetermined ways to commands from the executive function. For example, think of how the human body's organs respond to elevated temperatures of the environment. This property, which Gharajhedaghi calls "uni-mindedness," avoids conflicts between components. The uni-minded organization as organism is paternalistic: the executive function knows best. Program management examples of this model in history include the Manhattan Project, in which Robert Oppenheimer was [17] the executive function, and the strong chief-engineer organization used by Toyota for product development. In each new Toyota project the subordinate parts of the system, the automotive engineering groups, are disciplined to follow the company's tradeoff curves, which contain the accumulated knowledge and best practices—the corporate memory—of the company [18]. Conflicts between performing engineering groups are settled by the chief engineer, a person who has the respect of the organization by virtue of a long history in automotive engineering. By following this regimen, Toyota claims that its development process rarely misses milestones. But to say that Toyota fits the model of organization as organism ignores the social organization and culture that pervade Toyota and are believed to be responsible for the success of the firm [19].

Thinking about organizations as brains yields insights into how an organization learns and gains intelligence. The prize, in Morgan's words, is "learning how intelligence can be distributed throughout an enterprise and how the power of information technology can be used to develop decentralized modes of organization that are simultaneously global and local". Morgan divides this view into three, interconnected, components: organizations as information processing brains; organizations as complex learning systems; and organizations as holographic systems that combine centralized and decentralized characteristics. The organization as an information-processing brain plays on the observation that organizations are at once information systems, communication systems and decision-making systems [20]. Research to help organizations make more rational decisions has created the "big data" learning concept that increases an organization's capacity for managing its customers and its inbound and outbound logistics functions as well as its supporting functions such as finance, engineering and legal [21]. Organizations have also learned how to exploit communications and networking technology to organize "virtually," whereby functions and services can be provided in globally distributed locations that appear to be collocated. Apple's branding "designed in Cupertino, California, and manufactured in China" exemplifies a modern virtual organization [22]. The learning twist to this model is that Chinese engineers are empowered to change an Apple design to improve its manufacturability. Thus, Apple has learned how to transform a centralized, co-located organization (e.g., Lockheed's "Skunk Works" [23], famous for the development of complex, high-performance products) into a globally distributed organization that, moreover, has large production rates of high-performance products. In 1989 at the start of DARPA's Initiative in CE [24], the vision of a virtual meeting supported by engineering design and "virtual factory" toolsets and enabled by then-current client–server computer technology seemed far-fetched [25]. Indeed, rudimentary demonstrations of these concepts proved difficult and were hard won. But by the late 1990s, with rapid evolution of the internet and related web technologies and interface and communication standards, the virtual organization and its enabling of distributed CE teams had become a reality. From a program management perspective, total product life-cycle management (PLM) that incorporated program learning into standardized work of best known practices was also possible with extant information technology and product application tools [26]. Today, Web 2.0 tools are evolving to higher-order tools with added functionality [21].

The last facet of the metaphor of organizations as brains examines learning organizations and how organizations can "learn to learn." Here we adopt Senge's definition that learning is expanding the ability to produce the results we truly want in life [3]. Morgan starts with two modes of learning, single-loop learning and double-loop learning, that were identified by Argyris [27]. Single-loop learning occurs when an organization senses a departure from an operating norm; it then instigates problem solving to discover the cause and implements corrective feedback to restore operation to the norm. In double-loop learning, the organization questions, through an in-depth root causes analysis, whether the operating norm itself is relevant. If the operating norm is deficient or not relevant, the organization will devise, implement and track the effectiveness of a permanent, mistake-proof solution. For example, returns to the US OEM from Asian chip manufacturers of a robotic vision device for inspecting chips were increasing. It didn't take the OEM's troubleshooting group long to discover that the common cause was dirty optical components. So the group cleaned all optical elements, re-calibrated the inspection machines to original design settings, and returned the machines to the manufacturers. However, this exercise in single-loop learning didn't staunch the flow of returns. Probing deeper by applying double-loop learning, an outside consultant discovered that the machine never had a design specification ("norm") for either dust protection or the dust content of the operating environment. This finding led to solutions for improved sealing of the machine's enclosure and for controlling the dust level in customers' factories, not to mention the addition of a design specification for new machines.

A dimension of learning-to-learn is Nonaka's work on the learning cycle, through which tacit knowledge (i.e., skills, experience and craft) is transformed into explicit knowledge that is documented and imbedded in readable instructions and tools that are usable by the broader workforce population [4] (an example of explicit knowledge are Toyota's tradeoff curves). Standardized work is often misunderstood as a collection of permanent, "set-it-and-forget-it" practices. Unless standardized work represents an organic body of the best current practices, it will lead to a moribund organization.

Another perspective of organization produced the sociocultural model. It has evolved from a systems inquiry of the organization as mechanical system, but with interdependent variables, and came later in the twentieth century with the application of the mathematics of Operations Research (OR) to find optimal solutions to networks of interdependent variables. Academics like Ackoff and Forrester (systems dynamic modeling) [28] and practitioners like McNamara [29] contributed to this mode of inquiry. Eventually Ackoff [30] would renounce the OR approach because it didn't consider that the parts in a social system have a choice.

In its place Ackoff advocated a new generation of systems thinking, the sociocultural model of organization, whereby the organization is distinguished as being a multi-minded system. The sociocultural model holds that an organization is comprised of members who have both choice and purpose [8]. Gharajedaghi writes "… the purpose of an organization is to serve the purposes of its members while also serving the purposes of its environment." Members are bound together by a common set of values (or justifications and purposes) and beliefs, in how the organization and their careers have succeeded in the past. These intangible values and beliefs, together with tangible artifacts (e.g., results, policies, performance measures, evaluation criteria, rewards and consequences, technology), constitute the shared culture of the organization [31]. Taken together the three components of culture (shared values, artifacts, deeply-held beliefs and assumptions) motivate the view that culture is the operating system of an organization. By contrast with the mechanical model of organization, members of the socio-cultural organization have ownership of the organization's activities, including the execution of programs, because they are included in strategy, design and planning decisions.

We place the sociotechnical system as a subset of this view, whereby the coordination of the technical elements (structure, process, job design, technology…) and of the organization with its social elements (both members, partners, and society) are essential to success. Today's vexing, global problems of climate change, economy, energy, security, mobility, health care, and peace are all examples of complex sociotechnical systems. Integration of component parts in configurations that best serve an organization's and its environment's purposes is an ongoing activity. In Project Design, we seek to represent dependencies and integrate among three components: the product or system breakdown structure (as guided by its constituents' needs), the process ("work") breakdown structure (as guided by the activity requirements), and the organization or team breakdown structure (as determined by resources and cultural norms).

The role of culture in program management is powerful, as has been discovered, for example, in investigations of major accidents such as NASA's Challenger and Columbia shuttle explosions and BP's Horizon Oil Rig blow out. In both organizations the drive for performance (a cultural norm) interfered with situational awareness and safety during program execution. These accidents also highlighted the absence of adequate risk management practices, even though risk management has [32] long been a principal tool of program management.

A deeper view of risk [33, 34] in program management uncovers humans' lack of training in making estimates, including estimating risk in the face of uncertainty [32]. Starting with the unsung article "Managing Project Uncertainty: From Variation to Chaos" [35], and progressing to scholarly works on the flaw of averages [36] and black swans, we find that contemporary notions of uncertainty and how to manage it have outstripped the classic risk "cube" (actually, a two-dimensional matrix) and related tools that are still in widespread use.

Within the umbrella of organizational thinking, program management has evolved as a business process that is commonly found allied with policies and procedures for new product development. Its improvements have been slow, because it is here that the weight of an organization's entrenched culture and its pride as an innovator are felt. For organizations that haven't put in place a Program Management team (such as a PMO) with a discipline of learning and continuous improvement (kaizen), leadership delegates a program manager to an important program with the expectation that the job will either get done as planned or a "better" program manager will be impressed into the assignment. Rather than being a source of help to the program manager, leadership assumes a command and control management style under which learning anxiety thrives [31]. If a program fails, the program manager is blamed.

The quality and productivity movements (Total Quality Management, Six Sigma and Lean Thinking) that swept through organizations in the last three or so decades were hard to apply to program management because of its business process nature and the fact that the teaching examples were geared toward more transparent manufacturing processes. Only relatively recently have works of "translation" come to light for how to apply lean thinking to product and process development [18, 37].

3.5 Evolution of Systems Thinking

Gharajedaghi defines systems thinking as the antithesis of analysis: "(Systems thinking) puts the system in the context of the larger environment of which it is a part and studies the role it plays in the larger whole" [8]. He observes that systems thinking is several centuries newer than the analytical approach but has already undergone three distinct generations of change:

- First generation (operations research)—interdependency in the context of mechanical (deterministic) systems
- Second generation (cybernetics and open systems)—interdependency and self-organization in the context of living systems
- Third generation (interactive design)—interdependency, self-organization and purpose and choice in the context of sociocultural systems.

One can see the parallels between the evolutions of systems thinking and organization models, from mechanical, to biological to sociocultural. A distinguishing feature of these models is the nature of the bond between system components (members). In mechanical systems, the bonds are energy related and governed by the laws of nature; in biological and in sociocultural systems, the bonds are information-oriented.

The notion of self-organization means that the system moves toward a pre-defined order, which is an internally-held, shared image of what the system desires to become. Gharajedaghi argues that this image or blueprint is DNA for a biological system and culture for a sociocultural system. He makes a strong point that the shared image of culture defines default values for future decisions and that the persistence of these default values makes it difficult to create change in sociocultural systems.

To complement his discourse on system thinking, Gharajedaghi proposes that civilization is the emergent outcome of the interaction between culture and technology. He likens culture to the software of an operating system, while technology is related to hardware and tools [8]. The evolutionary step to the third generation of systems thinking is provided by Ackoff's idealized design process [38] for interactive design—a systems thinking methodology for managing the interdependencies between culture and technology. A sub-theme in third generation systems thinking is how to manage an uncertain future.

The process of interactive design, which Ackoff calls idealized design, guides members of an organization in the creation of a plan for what they want the organization to be at the present. Since the plan was designed by thinking backward from where they want the organization to be to where it is now, idealized design also helps prepare the organization for success in an unknowable future.

Another example of third-generation systems thinking is provided by the work of the Society for Organizational Learning (SoL) and its team members. Their first exposition appeared in the book *Presence: Human Purpose and the Field of the Future* [39], followed by a deeper exploration of the underlying theory, called Theory U: The Social Technology of Presencing [40].

3.6 Force Fields

In setting the context for this discussion, we have provided a small taste of background concepts for program management and systems thinking. To summarize we prepared the force-field analysis (Fig. 3.1), which shows those concepts that are

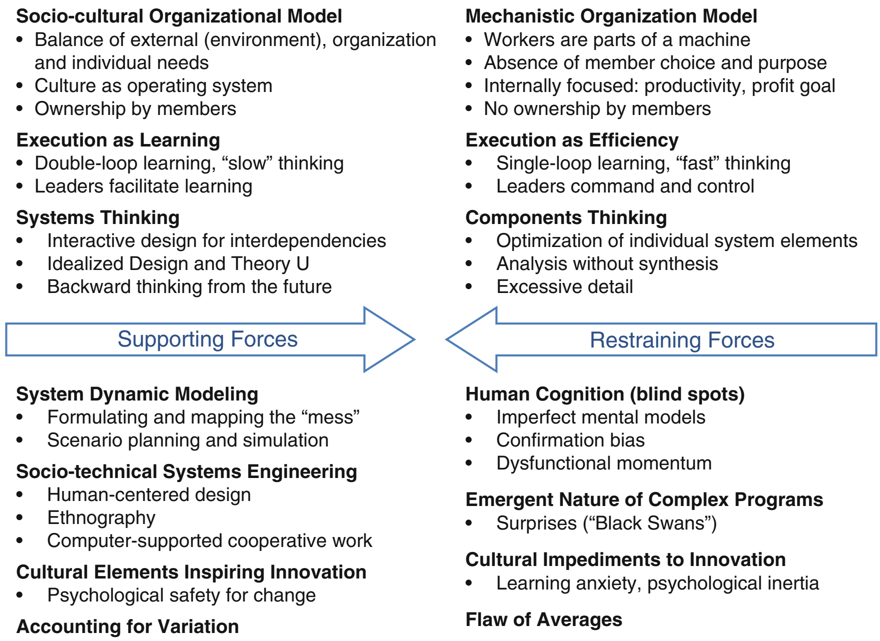


Fig. 3.1 Supporting and restraining forces for systems thinking in engineering programs

reinforcing effective program management in today's dynamic and global business environment, and those that may be retarding improvement.

These forces generate stress in day to day engineering programs, with some arguing that well-worn processes from past success should be followed, while other search for new "silver bullet" processes to replace them. Indeed, unless an organization engages in deeper levels of learning, the mental models and cultural habits of the past will continue to imprint on program management of new endeavors. The nature of new programs as dynamic sociotechnical systems is in contrast to the centralized, structured, and mechanistic work systems dominant a century ago. Continuing to follow old frameworks for new conditions will continue to cause unexpected and undesirable results.

3.7 Conclusion

Leveraging systems thinking, we have sought to better represent and forecast engineering programs as sociotechnical systems. We argue with others that the complexities of real world engineering programs are understood better through consideration of emergent outcomes driven by the dynamic interaction of technical and social characteristics. We have been deploying methods and tools based on this thinking in the field for over a decade. We build models of projects, simulate to forecast likely outcomes, and lead cross-functional teams to explore and converge across scenarios in planning workshops. Rather than finding a single optimal process, which assumes stability and absence of real-world uncertainty in cost, schedule, and quality, we capture the characteristics of the project that will allow insights into feasibility, value, and a trade space of likely outcomes. We refer to this integrated approach as *Project Design*, further introduced in Chap. 8.

References

1. Kahneman D (2011) Thinking, fast and slow. Macmillan, New York
2. Ackoff R, Rovin S (2003) Redesigning society. Stanford University Press, Stanford
3. Senge P (2006) The fifth discipline: the art and practice of the learning organization. Random House Digital, New York
4. Nonaka I, Takeuchi H (1995) The knowledge-creating company: how Japanese companies create the dynamics of innovation. Oxford University Press, New York
5. Edmondson A (2008) The competitive imperative of learning. Harvard Bus Rev 86(7/8):60–69
6. Kendall HP (1912) Unsystematized, systematized, and scientific management. In: Addresses and discussions at the conference on scientific management held at Dartmouth College
7. Clark W (1922) The Gantt chart: a working tool of management. The Ronald Press Company, New York
8. Gharajedaghi J (2011) Systems thinking: managing chaos and complexity, 3rd edn. Elsevier, Burlington
9. Morgan G (1997) Images of organization. SAGE Publications, Thousand Oaks
10. Gantt H (1913) Work, wages, and profits. Engineering Magazine Co., New York
11. Fayol H (1949) Industrial and general management. Pitman, London
12. McMillan E (2008) Complexity, management and the dynamics of change: challenges for practice. Routledge, Abingdon
13. Weber M (1964) The theory of social and economic organization. Collier-Macmillan, London
14. Burns T, Stalker G (1961) The management of innovation. University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship
15. Simon H (1996) The sciences of the artificial. MIT Press, Cambridge
16. Burton RM, Obel B (1998) Strategic organizational diagnosis and design: developing theory for application. Kluwer Acadeic Publishers, Norwell
17. Rhodes R (1986) The making of the atomic bomb. Simon & Schuster, New York
18. Kennedy MN, Ward A (2003) Product development for the lean enterprise. Oaklea Press, Richmond
19. Liker J (2004) The Toyota way. McGraw-Hill, New York
20. Morgan G (1997) Learning and self organization: organizations as brains. In: Images of organization–executive edition, SAGE Publications, Thousand Oaks
21. Mcafee A (2006) Enterprise 2.0: the dawn of emergent collaboration. Manage Technol Innov 47(3):21−28
22. Duhigg C, Bradsher K (2012) Apple, America and a squeezed middle class. The New York Times, New York, 21 Jan 2012
23. Ziemke M, Spann M (1993) Concurrent engineering's roots in the World War II era. In: Concurrent engineering. Springer, Berlin, pp 24–41
24. Reddy R, Wood R, Cleetus K (1991) Concurrent engineering: the DARPA initiative: encouraging new industrial practices. IEEE Spectr 28(7):26–27

25. Lucas M, Qudsi U, Brence P, Garber W, Michaels R, Shapiro D, Peck A, Croom P, Garner H (1992) DARPA initiative in concurrent engineering (DICE) phase 4. In: Electronics pilot project final technical report, DTIC Document
26. Immonen A, Saaksvuori A (2005) Product lifecycle management, 2nd edn. Springer, Berlin
27. Argyris C (1977) Double loop learning in organizations. Harvard Bus Rev 55(5):115–125
28. Forrester J (1961) Industrial dynamics, vol 2. MIT Press, Cambridge
29. Bunkley N (2007) J. Edward Lundy, 'Whiz Kid' at Ford Motor, Dies at 92. New York Times, New York, 06 Oct 2007
30. Ackoff R (1979) The future of operational research is past. J Oper Res Soc 30(2):93
31. Schein E (2009) The corporate culture survival guide. Wiley, San Francisco
32. Hubbard DW (2009) The failure of risk management: why it's broken and how to fix it. Wiley, Hoboken
33. NN (2001) Project management body of knowledge (PMBOK® GUIDE). Project Management Institute
34. Smith P, Merritt G (2002) Proactive risk management. Productivity Press, New York
35. De Meyer A, Loch C, Pich M (2002) From variation to chaos. MIT Sloan Manage Rev 43:60–67
36. Savage S (2012) The flaw of averages: why we underestimate risk in the face of uncertainty. Wiley, Hoboken
37. Ward A (2007) Lean product and process development. Lean Enterprise Institute, Cambridge
38. Ackoff R, Magidson J, Addison H (2006) Idealized design: how to dissolve tomorrow's crisis… today. Pearson Prentice Hall, Upper Saddle River
39. Senge P, Scharmer CO, Jaworski J, Flowers BS (2004) Presence: human purpose and the field of the future. Society for Organizational Learning, Cambridge
40. Scharmer CO (2009) Theory U: learning from the future as it emerges. Berrett-Koehler Publishers, San Francisco

Chapter 4
Technology Foundations

Michael Sobolewski

Abstract The chapter focuses on the underlying concepts of concurrent engineering technology from the point of view of concurrent process expression and its actualization. It lays out the evolution of computing platforms and networking complexity that constitute the foundation of every distributed information system required for concurrent engineering. Network integration is the working foundation for computer-based approaches to concurrent engineering. Therefore, an architecture of a concurrent engineering system is presented from the point of view of evolving methodologies of remote method invocation. Within the architecture the integration of concurrent distributed processes, humans, tools and methods to form a transdisciplinary concurrent engineering environment for the development of products is presented in the context of cooperating processes and their actualization. To work effectively in large, distributed environments, concurrent engineering teams need a service-oriented programming methodology along with a common design process, domain-independent representations of designs, and general criteria for decision making. Evolving domain-specific languages (DSLs) and service-oriented platforms reflect the complexity of computing problems we are facing in transdisciplinary concurrent engineering processes. An architecture of a service-oriented computing environment (SORCER) is described with a service-oriented programing and a coherent operating system for transdisciplinary large-scale computing.

Keywords Concurrent engineering · Metacomputing · Transdisciplinary computing · Service-oriented architectures · Var-modeling · Var-oriented programming · Exertion-oriented programming

M. Sobolewski (✉)
United States Air Force Research Laboratory, WPAFB, Dayton, OH 45433, USA
e-mail: sobol@sorcersoft.org

M. Sobolewski
Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_4

4.1 Introduction

Concurrent engineering, as the name suggests, is the approach of doing all necessary activities at the same time [1]. It is the unison of all facets of the product life cycle to minimize modifications in a prototype, i.e., to decrease design iterations performed during product development.

Concurrent Engineering (henceforth CE) is characterized by a focus on customer requirements. Moreover, it embodies the belief that quality is built into the product, and that it (quality) is a result of continuous improvement of a process. This concept is not new; in fact, the approach is quite similar to the "tiger team" approach characteristic of small organizations. The "tiger team" essentially is a small group of people working closely for a common endeavor, which might be product development. The magnitude of the problem is usually small with few conflicting constraints. The approach works well for small organizations; however, in large organizations the technique needs to be modified and restructured. It is here that CE comes into picture. CE envisages translating the "tiger team" concept to big organizations and such "tiger teams" will work with a unified product concept. Because team members can be at geographically different, networked locations, this requires far-reaching changes in the work culture, ethical values and information technology (IT), and a distributed infrastructure of the organization.

Commonplace design activities involve sequential information transfer from "concept designers" to "design finishers". When design activities are finished, the people involved in them get detached from the design chain. Thus, the people involved in earlier design phases do not interact with people in the later stages. A natural consequence of this procedure is that errors go on propagating themselves down the chain and are usually detected at a stage where rectifications/modifications become both costly and undesirable. The philosophy of continuous improvement implies changes at the initial stages with the aim of minimizing changes at later stages. To achieve this, it is imperative that strong communication exists between product developers of all stages and end-users.

Integrated, parallel, product and process design is the key to concurrent design. The CE approach as opposed to the sequential approach advocates such a parallel design effort. The objective is to ensure that serious errors don't go undetected and that the design intent is fully captured. The above-mentioned integrated design process should have the following features:

1. There must be strong information sharing system, thus enabling design teams to have access to all corporate facilities as well as work done by individual teams.
2. Any design process is necessarily an iterative process requiring successive redesigns and modifications. The CE process should ensure that the effects of a change incorporated by one team on other design aspects are automatically analyzed. Moreover, the affected functional units should be notified of the changes.

3. The CE process must facilitate an appropriate trade-off analysis leading to product-process design optimization. Conflicting requirements and constraint violations must be identified and concurrently resolved.
4. All relevant aspects of the design process must be recorded and documented for future reference.

The integration process discussed here is *transdisciplinary* and *strategic* integration in an organization. This binds the various functional discipline areas as engineering, support, manufacturing, logistics, etc., in the organization for greater efficiency of the whole venture. Strategic integration focuses on a company's business strategy. This strategy should tie decision making and other organizational policies together with the objective of realizing total quality management. Logistic integration is basically close coordination of the manufacturer with its customers and suppliers for cutting down logistic problems.

The realization of functional integration requires a versatile and a flexible *information management system* within the organization. The system must be domain independent and adaptable to both large and small industrial enterprises. Some essential and desirable characteristics of this system are listed below:

1. The system should be adaptable to the needs of the specific organization. It should be generic and at the same time modifiable to the requirements of the enterprise.
2. It should have a diverse repository of organized knowledge which is easily accessible across the spectrum of product life-cycle disciplines.
3. There must be an intelligent information distribution system which could provide information on a "need to know" and/or user-specified basis.
4. It should have facility of interfacing with software tools and application databases existing in user's organization.
5. The system should be capable of making the whole design team cognizant of the modifications done by a sub-group. In addition, it should have the ability to appraise the impacts of the modifications in a global manner, i.e., on all other design activities.
6. Even though most of the activities should be automated, there must be provision for human intervention at every stage. Furthermore, a manual bypass alternative for autonomous activities should be provided.
7. The system must support progressive refinement of product and process development from "design initiation" to the "design finalization" stage.
8. There must be tools permitting rapid prototyping and testing, therefore paving the way for commercial production.

The information management system can be visualized as the environment that allows expression of concurrent processes and their actualization. While process expression requires relevant domain-specific languages (DSLs) their actualization requires three basic blocks: data architecture (domain), distributed management framework (operating system), and software services (domain-specific processor).

Understanding the principles that run across process expressions and appreciating which language features and computing platforms are best suited for which type of process, bring these process expressions to useful life. No matter how complex and polished the individual process operations (software services) are, it is often the quality of the distributed operating system that determines the power of the concurrent engineering environments.

This chapter gives insight in the technological foundations of CE. In the Sect. 4.2 the basic terms are described from their historical perspective. Section 4.3 gives overview of rising complexity of computing. In the following Sect. 4.4 service platforms are classified in three categories and compared. Typical use case is described in Sect. 4.5, followed by conclusions and outlook (Sect. 4.6).

4.2 Background

Markov tried to consolidate all work of others on effective computability. He has introduced the term of algorithm in his 1954 book Teoriya Algorifmov [2]. The term was not used by any mathematician before him and reflects a limiting definition of what constitutes an acceptable solution to a mathematical problem:

> In mathematics, "algorithm" is commonly understood to be an exact prescription, defining a computational process, leading from various initial data to the desired result [2].

The important keyword in the Markov definition is "computational process" and the "algorithm" can be seen as a way to mathematically express the process. The mathematical view of process expression has limited computing science to the class of processes expressed by algorithms. The following definition of an algorithm (consistent with Hilbert's proposal of 1920) is typical:

1. An algorithm must be a step-by-step sequence of operations.
2. Each operation must be precisely defined.
3. An algorithm must terminate in a finite number of steps.
4. An algorithm must efficiently yield a correct solution.
5. An algorithm must be deterministic in that, given the same input, it will always yield the same solution.

From experience in concurrent engineering since 1989 it becomes obvious that in computing science the common thread in all computing disciplines is process expression; that is not limited to algorithm or actualization of process expression by a single computer. Several process expressions have been defined. Below a list of known process expression and actualization solutions is presented:

1. An architecture is an expression of a continuously acting process to interpret symbolically expressed processes.
2. A user interface is an expression of an interactive human-machine process.

3. A mogram (which can be program or model, or both) is an expression of a computing process [3].
4. A mogramming (programming or modeling, or both) language is an environment within which to create symbolic process expressions (mograms).
5. A compiler is an expression of a process that translates between symbolic process expressions in different languages.
6. An operating system is an expression of a process that manages the interpretation of other process expressions.
7. A logic circuit is an actualization of a logical process.
8. A processor is an actualization of a process.
9. An application is an expression of the application process.
10. A computing platform is an expression of a runtime process defined by the triplet: domain—mogramming language, management—operating system, and carrier—processor.
11. A computer is an actualization of a computing platform.
12. A metamogram (metaprogram or metamodel, or both) is an expression of a metaprocess, as the process of processes.
13. A metamogramming language is an environment within which to create symbolic metaprocess expressions.
14. A metaoperating system is an expression of a process that manages the interpretation of other metaprocess expressions.
15. A metaprocessor is an actualization of the metaprocess on the aggregation of distinct computers working together so that to the user it looks and operates like a single processor.
16. A metacomputing platform is an expression of a runtime process defined by its metamogramming language, metaoperating system, and metaprocessor.
17. A metacomputer is an actualization of a metacomputing platform.
18. Computer science is the science of process expression.
19. Computer engineering is the science of process actualization.
20. Software engineering is an expression of a reliable development process within which to create program design, its implementation, and all related documents.
21. An information system is an expression of an efficient process to retrieve, store, and transmit information.
22. IT is an expression of a reliable (24/7) process to maintain and manage computing and data assets to meet current and expected user needs.
23. Artificial intelligence is an expression of integrated processes to reason, learn, plan, communicate knowledge, and perceive the world to move and manipulate objects.
24. Concurrent engineering is an expression of concurrent integrated processes to develop complex products.
25. Cloud computing is an expression of a consolidated process using virtualized (guest) platforms on native (host) platforms with related software as services running on these host and guest platforms.

Obviously, there is an essential overlap between the domains of mathematics and computer science, but the core concerns with the nature of process expression itself is usually ignored in mathematics as mathematicians are concerned with the nature of behavior of a process independent of how that process is expressed. By contrast, computer science is mainly concerned with the nature of the expression of processes independent of its behavior. That became obvious in 1990s when computer science was redefined as one of many disciplines of computing science (see Fig. 4.1):

I. Computer Engineering (CE)
II. Computer Science (CS)
III. Software Engineering (SE)
IV. Information Technology (IT)
V. Information Systems (IS)
VI. Concurrent Engineering (CCE).

A comprehensive definition of concurrent engineering is given in the IDA (Institute for Defense Analysis) report on concurrent engineering (see Chap. 2) [4]:

> Concurrent engineering is the systematic approach to the integrated, concurrent design of products and related processes including manufacture and support. This approach is to cause the developers, from the outset, to consider all the elements of product life-cycle from conception through disposal including quality, cost, schedule and user requirement.

The concurrent engineering method is still a relatively new design management system, but has had the opportunity to mature in recent years and to become a well-defined systems approach towards optimizing engineering design cycles. One of the most important reasons for the huge success of concurrent engineering is that by definition it redefines the basic design process structure that was commonplace for

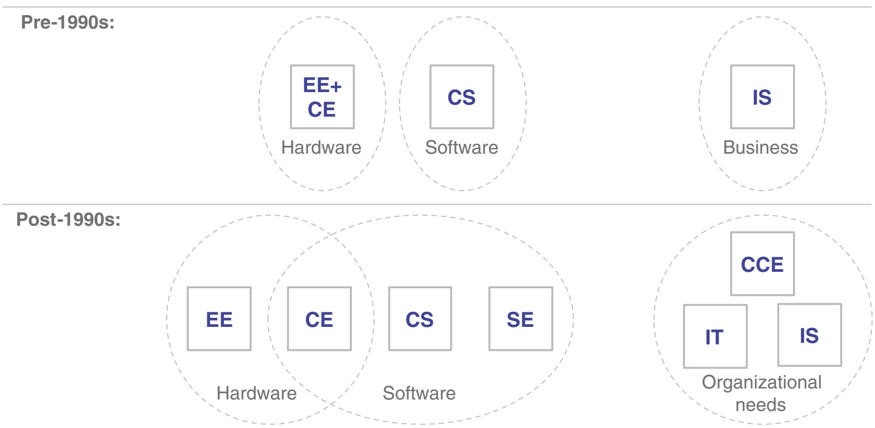


Fig. 4.1 More computing disciplines post 1990s; computer science becoming the science of process expression. Concurrent engineering (*CCE*) initially funded by DICE/DARPA in 1989 has become a new discipline of computing science and engineering to address organizational needs

decades. This was a structure based on a sequential design flow. Concurrent engineering significantly modifies this outdated method and instead opts to use what has been termed an iterative or integrated development method based on concurrent (non-algorithmic) process expression in socio-technical systems (see Chap. 8).

The notion of algorithm separates all process expressions into algorithm and non-algorithm, but what purpose does it serve to know that one program is an acceptable mathematical solution and another is not? If the same process can be expressed by an algorithm and by a non-algorithm then which expression is better? Does determining whether or not a given expression is an acceptable mathematical solution implies a better computer system or helps in writing a better program? In fact, important process expressions do not qualify as Markov algorithms. A process defined by a neural network or every logic circuit is not a sequence of operations—it is not a process expression of the mathematical type. An operating system does not to have to terminate or yield a singular solution. It is not deterministic as it receives uncoordinated inputs from the outside world. Any simulation process with random inputs is not an algorithm. No program with a bug can be an algorithm as well as a concurrent program that does not satisfy the concept of sequential behavior. These and other facts have forced computer scientists to patch the concept of algorithm with multiple revisions and redefinitions of algorithm (nondeterministic algorithm, semi-algorithm, probabilistically good algorithm, random algorithm, infinite algorithm, etc.). Thus, the notion of mathematical algorithm simply does not provide a conceptual framework for questions that mostly computer scientists are concerned with nowadays.

Research in knowledge-based process expression at the Concurrent Engineering Research Center, West Virginia University (CERC/WVU, 1989–1994) has resulted in the DICEtalk platform [5]. Later, further work at the GE Global Research Center (GE GRC 1994–2002), Texas Tech University (TTU, 2002–2009), and Air Force Research Lab (AFRLWPAFB, 2006–present) has been focused on large-scale distributed process expression actualized by service-oriented platforms. Computing concepts and related results have been verified practically in large-scale real world systems for concurrent engineering. For the description of programming and distributed computing relevant to concurrent engineering platforms presented in this chapter we have taken into account most of the relevant papers published before in this area.

4.3 Complexity of Computing Systems

The concurrent engineering challenge is not to get lost in the continuously increasing complexities of products, their development processes and computing platforms. In this section we consider evolving complexity of programming languages and platforms with the complexity of remote procedure calls (RPC) directly related to network programming. Then service-oriented concepts are described with a meta-modeling architecture with convergence of three computing platforms for service-oriented mogramming.

Thinking more explicitly about programming languages (languages for humans) instead of software languages (languages for computers) may be our best tool for dealing with real-world complexity. Understanding the principles that run across process expressions and appreciating which language features and related computing platforms are best suited for which type of process, bring these process expressions to useful life. No matter how complex and polished individual process operations are (tools, applications, and utilities), it is often the abstraction and quality of the operating system and underlying network processor that determine the power of the computing system.

4.3.1 Meta-computing

The term "metacomputing" was coined around 1987 by NCSA Director, Larry Smarr: "The metacomputer is, simply put, a collection of computers held together by state-of-the-art technology and balanced so that, to the individual user, it looks and acts like a single computer. The constituent parts of the resulting metacomputer could be housed locally, or distributed between buildings, even continents."

From the very beginning of networked computing, the desire existed to develop protocols and methods that facilitate the ability of people and automated processes across different computers to share resources and information across different computing nodes in an optimized way. As ARPANET [6] began through the involvement of the NSF [7, 8] to evolve into the Internet for general use, the steady stream of ideas became a flood of techniques to submit, control, and schedule jobs across distributed systems [9, 10]. The latest of these ideas are the grid [11] and cloud [12], intended to be used by a wide variety of different users in a non-hierarchical manner to provide access to powerful aggregates of resources. Grids and clouds, in the ideal, are intended to be accessed for computation, data storage and distribution, and visualization and display, among other applications without regard for the specific nature of the hardware and underlying operating systems on the resources on which these jobs (executable files) are carried out. While a grid is focused on computing resource utilization, clouds are focused on platform virtualization in computer networks. In general, grid and cloud computing is client-server computing that abstract the details of the server away—one requests a service (resource), not a specific server (machine). However, both terms are vague from the point of view of computing architectures and programming models and refer to "everything that we already do" with executable files and client-server architectures.

As we reach adolescence in the Internet era we are facing the dawn of the metacomputing era, an era that will be marked not by PCs, workstations, and servers, but by computational capability that is embedded in all things around us containing service providers. These service providers just consume services and provide services from and to each other respectively. Applications are increasingly moving to the network—self-aware, autonomic networks that are always fully functional. Service providers hosted by service objects (service containers) implement instructions of the virtual service processor (meta-processor). The meta-processor, with the help of its operating system, carries access to applications, tools, and utilities, i.e., programs as the instructions (services) of the meta-processor (while a processor is executing native machine instructions of executable codes). Services can collaborate with each other dynamically to provide aggregated services that execute a program collaborating with other component programs remotely and/or locally. Thus, a metacomputer is a collection of computers and devices connected by communication channels that facilitates distributed inter-process communications between users and allows users to share resources with other users.

Therefore, every meta-computer requires a computing platform that allows software to run utilizing multiple component computing platforms that communicate through a computer network. Different distributed platforms can be distinguished along with corresponding meta-processors—virtual organizations of computing nodes.

4.3.2 Programming and Platform Complexity

The functionality of a computing platform depends on its operating system and its programming environment. Not every computing environment supports a complete platform. Each of them has a kind of programming environment but not each has an adequate operating system. For example, the first computer ENIAC did not have an operating system and it was programmed with switches and cables. In order to run a new program, ENIAC needed not only a new program to be entered manually using switches but also to be rewired. It took us the past half-century to move from programming environments with cables and switches, via perforated tapes, punch cards, to executable files and scripts to be easily created within integrated development environments (IDE) by software developers. The platforms and related programming models have evolved as process expression from the sequential process expression actualized on a single computer to the concurrent process expression activated on multiple computers. An evolution in process expression introduces new platform benefits but at the same time introduces additional programming complexity that operating systems have to deal with. We can distinguish eight quantum leaps in process expression and their related architectures:

1. Sequential programming (e.g. stored-program on digital computer)
2. Multi-threaded programming (e.g. Java platform)
3. Multi-process programming (e.g. Unix platform)
4. Multi-machine-process programming (e.g. computer network)
5. Knowledge-based programming (e.g. DICEtalk)
6. Service-protocol oriented programming (e.g. Web and Grid services)
7. Service-object oriented programming (e.g. Jini)
8. Federated service-object oriented programming (e.g. SORCER)

One of the key elements of each distributed platform is communication between computer nodes and management of reliable network connections. All service-driven platforms are usually focused on communication protocols and service execution using a form of network middleware. However, most do not have a service-oriented operating system that deals with efficient management of services as platform commands, front-end programming (service scripting), and reliable runtime networking.

4.3.3 Generations of Remote Procedure Call

Socket-based communication forces us to design distributed applications using a read/write (input/output communication) interface, which is not how we generally design non-distributed applications based on procedure call (request/response). In 1983, Birrell and Nelson devised the RPC [13], a mechanism to allow programs to call procedures on other hosts. So far, six RPC generations can be distinguished:

1. First generation RPC—Sun RPC (ONC RPC) and DCE RPC, which are language, architecture, and OS independent;
2. Second generation RPC—CORBA [14] and Microsoft DCOM/OPC, which add distributed object support;
3. Third generation RPC—Java RMI is conceptually similar to the second generation but supports the semantics of object invocation in different address spaces that are built for Java only [15]. Java RMI fits cleanly into the language with no need for standardized data representation, external interface definition language, and with behavioural transfer that allows remote objects to perform operations that are determined at runtime;
4. Fourth generation RPC—next generation of Java RMI, Jini Extensible Remote Invocation—JERI—[16] with dynamic proxies, smart proxies, network security, and with dependency injection by defining exporters, end points, and security properties in configuration files;
5. Fifth generation RPC—Web/OGSA Services [17, 18] and the XML movement including Microsoft WCF/.NET;
6. Sixth generation RPC—Federated Method Invocation (FMI) [19] allows for concurrent invocations on multiple federating compute nodes in the Service-Oriented Computing Environment (SORCER) [20–25].

All RPC generations listed above are based on a form of service-oriented architecture (SOA). CORBA, RMI, and Web/OGSA service providers are object-oriented wrappers of network interfaces that hide object distribution and ignore the real nature of a network using classical object abstractions that encapsulate network connectivity by using existing network technologies. The fact that object-oriented languages are used to create corresponding object wrappers does not mean that distributed objects created this way have a great deal to do with object-oriented distributed programming.

Each platform and its programming language reflect a relevant abstraction, and usually the type and quality of the abstraction implies the complexity of problems we are able to solve. For example, a procedural language provides an abstraction of an underlying machine language. In the SORCER environment developed at Texas Tech University [19], a service provider is a remote object that accepts network requests to participate in a collaboration—a process by which service providers work together to seek solutions that reach beyond what any one of them could accomplish on their own. SORCER messaging is based on exertions, the service commands that encapsulate explicitly data, operations, and control strategy. An exertion can federate multiple service providers concurrently according to its control strategy by managing transparently all low-level Jini/JERI networking details.

The SORCER meta-computing environment adds an entirely new layer of abstraction to the practice of service-oriented computing: exertion-oriented (EO) programming. The EO programming makes a positive difference in service-oriented programming primarily through a new meta-computing platform as experienced in many large-scale projects including applications deployed at GE Global Research Center, GE Aviation, Air Force Research Lab (AFRL), and SORCER Lab. The new abstraction is about managing object-oriented distributed system complexity laid upon the complexity of the unreliable network of computers—the meta-computer.

An exertion submitted to the network dynamically binds to all relevant and currently available service providers in the network. The providers that dynamically participate in this invocation are collectively called the exertion federation. This federation is also called the exertion meta-processor since federating services are located on multiple computer nodes held together by the SORCER operating system (SOS) so that, to the requestor submitting the exertion, it looks and acts like a single processor.

The SORCER environment provides the means to create service-oriented mograms [25] and execute them using the SORCER runtime infrastructure. Exertions can be created using interactive user agents uploaded/downloaded on-the-fly to/from service providers. Using these interfaces, the user can create, execute, and monitor the execution of exertions within the SORCER platform. Exertions can be kept for later reuse, allowing the user to quickly create new scripts or EO programs on-the-fly in terms of existing exertions, usually kept for reuse.

SORCER is based on the evolution of concepts and lessons learned in the federated intelligent product environment (FIPER) project [26], a $21.5 million program founded by National Institute of Standards and Technology (NIST). Academic research on FMI, SOS, and EO programming was established at the SORCER Laboratory, TTU (2002–2009), where twenty-eight SORCER related research studies were performed. Currently, the SORCER Lab (http://sorcersoft.org) as the independent open source organization is focused on maturing the SORCER platform in collaboration with AFRL/WPAFB, SORCERsoft.com, and collaborating partners in China, and Russia.

4.3.4 SOA and Metamodeling Architecture

The SOA is a software architecture using loosely coupled software services that integrates them into a distributed computing system by means of service-oriented programming. Service providers in the SOA environment are made available as independent service components that can be accessed without a priori knowledge of their underlying platform or implementation. While the client–server architecture separates a client from a server, SOA introduces a third component, a service registry, as illustrated in Fig. 4.2 (the left chart). In SOA, the client is referred to as a service requestor and the server as a service provider which is responsible for deploying a service in the network, publishing its service to one or more registries, and allowing requestors to bind and execute the service. Providers advertise their availability in the network; registries intercept these announcements and collect published services. The requestor looks for a service by sending queries to registries and making selections from the available services. Requestors and providers can use discovery and join protocols [16] to locate registries and then publish or acquire services in the network.

We can distinguish the service object-oriented architecture (SOOA), where providers are network objects accepting remote invocations (call/response), from the service protocol-oriented architecture (SPOA), where a communication (read/write) protocol is fixed and known beforehand by the provider and requestor. Based on that protocol and a service description obtained from the service registry, the requestor can bind to the service provider by creating a proxy used for remote communication over the fixed protocol. In SPOA a service is usually identified by a name. If a service provider registers its service description by name, the requestors have to know the correct name of the service beforehand.

In SOOA, a proxy—an object implementing the same service interfaces (service types) as its service provider—is registered with the registries and it is always ready for use by requestors. Thus, in SOOA, the service provider publishes the proxy as the active surrogate object with a codebase annotation, e.g., URLs to the code defining proxy behavior (RMI and Jini ERI). In SPOA, by contrast, a passive

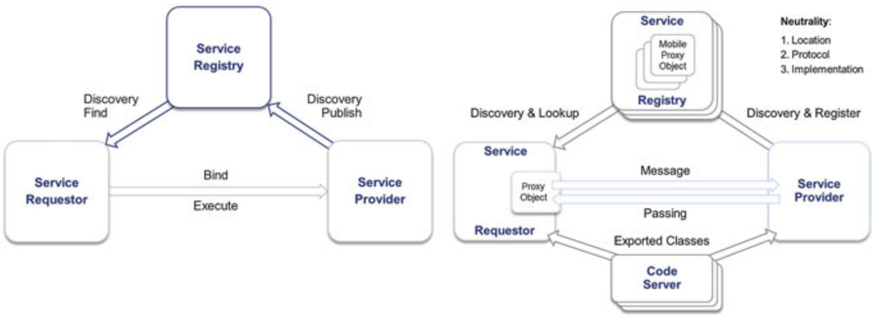


Fig. 4.2 SOA versus SOOA

service description is registered (e.g., an XML document in WSDL for Web/OGSA services, or an interface description in IDL for CORBA); the requestor then has to generate the proxy (a stub forwarding calls to a provider) based on a service description and the fixed communication protocol (e.g., SOAP in Web/OGSA services, IIOP in CORBA).

SPOA and SOOA differ in their method of discovering the service registry (see Fig. 4.2, the right chart). SORCER uses dynamic discovery protocols to locate available registries (lookup services) as defined in the Jini architecture [16].

Let us emphasize the major distinction between SOOA and SPOA; in SOOA, a proxy is created and always owned by the service provider, but in SPOA, the requestor creates and owns a proxy, which has to meet the requirements of the protocol that the provider and requestor agreed upon a priori. Thus, in SPOA the protocol is always a generic one, reduced to a common denominator—one size fits all—that leads to inefficient network communication in many cases. In SOOA, each provider can decide on the most efficient protocol(s) needed for a particular distributed application.

The first challenge of SORCER based on SOOA is to allow the end user not only to use the existing individual services as-is in the network, but also to create new compound services at runtime that are both globally and locally distributed federations of services. In other words, instead of invoking a single standard service in the network as-is, the computing environment should allow end users to create front-end complex service collaborations that become innovative new tools, applications, or utilities composed from the existing and new services at will.

The second challenge of the SORCER is to bring to front-end SO programming all existing programming styles seamlessly unified. Therefore, firstly the mogramming environment should be designed to express service collaborations functionally and procedurally as workflows. Secondly it should enable service interactions under control of the proper operating system to actualize the front-end services (exertions) using the domain-specific back-end heterogeneous services (service providers).

No matter how complex and polished the individual operations are, it is often the quality of the glue that determines the power of the distributed computing system. The SOS based on SOOA serves as the service-oriented glue for exertion-oriented mogramming [27]. It uses federated remote method invocation with location of service provider not explicitly specified in exertions [19]. A specialized infrastructure of distributed services supports discovery/join protocols for the SOS shell, federated file system, autonomic resource management, and the rendez-vous providers responsible for coordination of exertion federations. The infrastructure defines SORCER's service object-oriented distributed modularity, extensibility, and reuse of providers and exertions—key features of object-oriented distributed programming that are usually missing in SPOA programming environments. Object proxying with discovery/join protocols provides for comprehensive neutrality of provider protocol, location, and implementation that is missing in SPOA programming environments as well.

A meta-model is usually defined as a model of a model. What is meant is that where a model defines a system (instance), a meta-model (classifier) defines the model. In particular, the process expression in which other expressions are modeled is often called a meta-model. Note, that key modeling concepts are Classifier and Instance, and the ability to navigate from an instance to its classifier. This fundamental concept can be used to handle any number of layers (sometimes referred to as meta-levels).

The DMC meta-modeling architecture is based on the notion of the meta-model, also called the DMC platform or DMC-triplet, in the form: <Domain, Management, Carrier>. For example, a computing platform: <mograms, operating system, processor> is the model of the DMC-triplet. A language platform: <language expressions, grammar, language alphabet> is the model of the DMC triplet as well.

Therefore, a computing platform is a composition of a DSL, management of mogram execution, and the processor that provides the actualization of both the language and its management.

The SORCER meta-model is a kind of UML class diagram depicted in Fig. 4.3, where each "class" is a DMC-triplet. This meta-modeling architecture distinguishes three computing platforms (instruction set (IS), object-oriented, and service-oriented platforms) and three mogramming platforms [exertion-oriented programming (EOP), par-oriented modeling (POM), and var-oriented modeling (VOM)].

When dealing with meta-levels to define computing systems (process expression/actualization) there are at least two layers that always have to be taken into account: the process expression/actualization or the meta-model (abstract model); and the specification of the concrete process expression/actualization or the model (concrete model).

The domain-specific problems expressed with the higher expressive power of DSLs and adequate service-oriented syntax and semantics allow end users to be focused on their domain problems and solution. Unfortunately, the common service-based practice forces end users to learn back-end software-programming languages, development processes, and creating software for computers instead of writing easily understandable and modifiable front-end domain-specific mograms for and by others and themselves.

The SORCER reference architecture presented below in Fig. 4.5 is a model of the DMC meta-model depicted in Fig. 4.3. SORCER is the SOOA platform that introduces an innovative programming model and operating system to create easily service-oriented mograms that express complex domain-specific solutions. SORCER mograms—var-models, par-oriented, var-oriented, and EOP—with the abstraction of a virtual service-oriented processor, are executed in the network by the SOS. SORCER introduces a FMI [21] used in its SOOA [28] by the SOS. The SOS implements all features required for the SOO platform listed in Table 4.1.

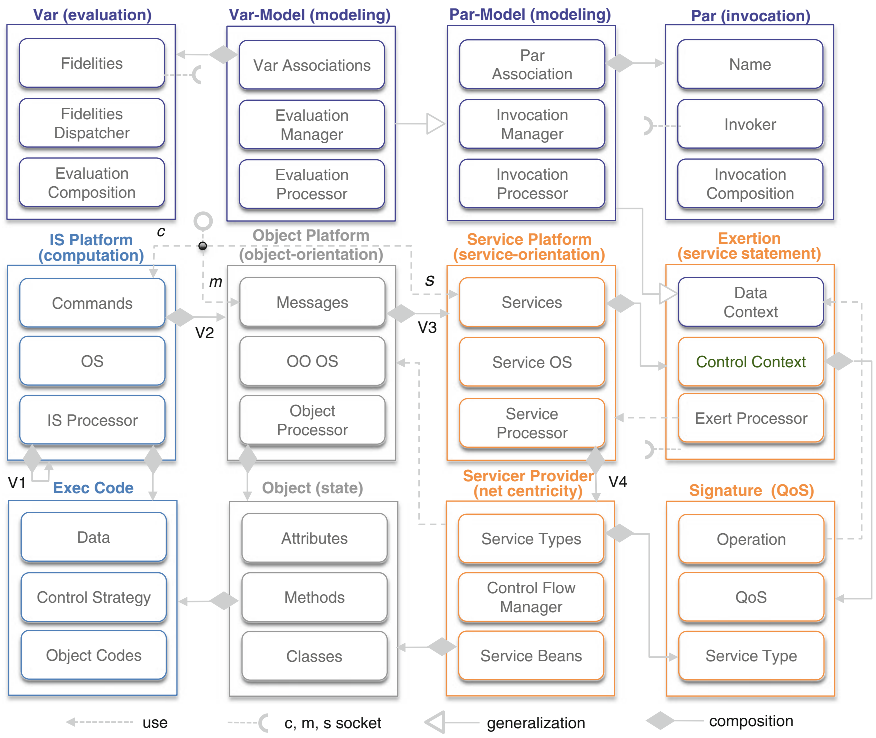


Fig. 4.3 The DMC meta-modeling architecture with platforms for: computation (instruction set), object-orientation, and service-orientation. Each platform is shown as the instance of DMC triplet with corresponding executable item (exec code, object, service provider). The top layer of the architecture shows elements of service-oriented modeling. That complements exertion-oriented programming. The service platform manages the service providers (virtualization of service cloud —*V*3/*V*4 that are autonomically provisioned by the SOS on virtualized computation/object platforms (*V*1/*V*2)

4.4 Service Platforms

In this section three basic categories of service platforms are distinguished: *multiple machine* (*MM*), *service protocol-oriented*, and *service-object-oriented platforms*. Then three types of SORCER-based platforms are described that are SOOA based specialization for *true service-oriented computing* (SORCER), *grid computing*, (SGrid) and at the *integration framework for intergrid computations* (iGrid).

Table 4.1 Quantum leaps in programming and platform complexity

Programming	Benefit	Lost benefit	Platform support
Sequential	Order		Batch processing before 1960s; from 1970s OS, e.g., UNIX with shell programming
Multi-threaded	Parallelism	Order	Thread management, e.g., Smalltalk, Java platform
Multi-process	SW isolation, safety	Execution context	OSs with pipes and sockets, e.g., UNIX with interprocess communication
Multi-machine	HW isolation, scalability	Global state, security	OSs with RPC and network tile systems, e.g., UNIX/NFS
Knowledge-base oriented	Ill-structured problem representation, logic (declarative) programming	Procedural execution	Knowledge representation handling, inference engine with procedural attachment, e.g., DICEtalk with percept knowledge representation
Service-protocol oriented	Service registry, code/resource location neutrality, implementation neutrality, platform neutrality	Trust, protocol neutrality as proxy is owned by the requestor, code/resource security due to dislocation	Service registry, protocol security, job scheduler, and virtual file system, e.g., CAMnet, CORBA, RMI, FIPER, Web Services, OSGA/Globus
Service-object oriented	Service-object spontaneity, code mobility, dynamic federations, registry location neutrality, protocol neutrality as proxy is owned by the provider, autonomic service-object/provider provisioning	Static service location, code security due to code mobility	Service interface types, object proxying, object registry, distributed events, transactions, leases, mobile code security, and disconnected operations, e.g., Jini/JERI, FMI/SORCER

4.4.1 Three Categories of Service Platforms

In meta-computing systems each service provider in the collaborative federation performs its services in an orchestrated workflow. Once the collaboration is complete, the federation dissolves and the providers disperse and seek other federations to join. The approach is service centric in which a service is defined as an independent self-sustaining entity—remote service provider—performing a specific network activity. These service providers have to be managed by a relevant operating system with commands for expressing interactions of providers in federations.

The reality at present, however, is that service-centric environments are still very difficult for most users to access, and that detailed and low-level programming must be carried out by the user through command line and script execution to carefully tailor jobs on each end to the resources on which they will run, or for the data structure that they will access. This produces frustration on the part of the user, delays in the adoption of service-oriented techniques, and a multiplicity of specialized "grid/cloud-aware" tools that are not, in fact, aware of each other which defeats the basic purpose of the grid/cloud.

Different platforms of meta-computers can be distinguished along with corresponding types of virtual service processors. For a meta-program, the control strategy is a plan for achieving the desired results by applying the platform operations (services) to the data in the required service collaboration and by leveraging the dynamically federating resources. We can distinguish three generic meta-computing platforms, which are described below. Meta-computing requires a relevant computing abstraction as well.

Procedural languages provide an abstraction of an underlying machine language. An executable file represents a computing program whose content is interpreted as a program on the underlying native processor. A command can be submitted to a *job* (*resource*) broker to execute a machine code in a particular way, e.g., by parallelizing and collocating it dynamically to the right processors in the network of compute resources. That can be done, for example, with the Nimrod-G grid resource broker scheduler or the Condor-G high-throughput scheduler [29]. Both rely on Globus/GRAM (Grid Resource Allocation and Management) protocol [18]. In this type of platform, called a *MM Platform* (see Fig. 4.4), executable files are moved around the network of compute nodes—the grid—to form virtual federations of required processors. This approach is reminiscent of batch processing in the era when operating systems were not yet fully developed. A series of programs ("jobs") was executed on a computer without human interaction or the possibility to

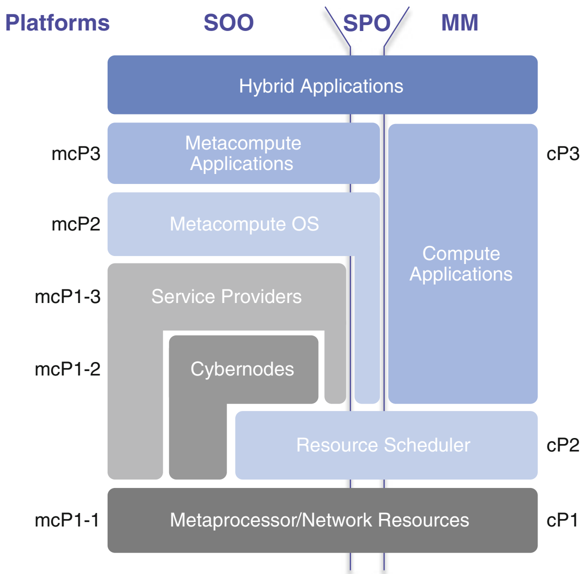


Fig. 4.4 Three types of platforms: MM-Platform (*cP1*, *cP2*, *cP3*), SPO-Platform in the middle between two *vertical lines* (*cP1*, *cP2*, *mcP2*, *mcP3*), and SOO-Platform (*mcP1-1*, *mcP1-2*, *mcP1-3*, *mcP2*, *mcP3*). A cybernode provides a lightweight dynamic service container (service object), turning heterogeneous compute resources into homogeneous services available to the meta-computing OS

view any results before the execution is complete. The UNIX operating system (OS) of the 1970s matured with the well understood architectures used still successfully nowadays.

We consider a true meta-program as the process expression of hierarchically organized collaboration of remote component programs. A meta-program is a program of programs such that its instructions correspond to dynamically bound service providers corresponding to applications, tools, and utilities in the network. Its service-oriented operating system makes decisions about where, when, and how to run these service providers. In other words, the meta-program manipulates other programs remotely and dynamically as its data. Nowadays the similar computing abstraction is usually applied to the program executing on a single computer as to the meta-program executing in the network of computers, even though the executing environments (platforms) are structurally completely different. Most so called service-oriented programs are still written using software languages such as FORTRAN, C, C++ (compiled into native processor code), Java, Smalltalk (compiled into intermediate code), and interpreted languages such as Perl and Python the way it usually works on a single host. The current trend is to have these programs and scripts define remote computational modules as *service providers*. However, most grid programs are developed using the same abstractions and, in principle, run the same way on the MM-Computer as on a single computer, for example using executable codes moved to the available computing nodes in the network.

The meta-computer based on Web Services or Grid Services can be considered as the SPO-Computer with SPO-Platform shown in Fig. 4.4. The SPO-Platform uses a SPO-Manager running usually on a Web Application Server. This type of meta-computing in concept is reduced practically to the client-server model with all drawbacks related to static network connections as discussed in Sect. 4.3.3. The Web Services model with a SPO-Manager that supports for example Business Process Execution Language (BPEL) [30] allows for deployment of service assembly on the application server that looks to the end user as a single server command—not a program for the service collaboration created by the end user. In this case the WSBPEL compliant WS engine is required on the application server, for example Apache ODE. The engine organizes web services' calls following a process description written in the BPEL XML grammar. BPEL's messaging facilities depend on the use of the Web Services Description Language (WSDL) to describe outgoing and incoming messages from web services as specified during the service deployment on the application server, that is not available to end users.

We can distinguish three types of computing platforms depending on the nature of network operating system or middleware (see Fig. 4.4): the MM-Platform with computing layers cP1, cP2 and cP3; the SPO-Platform with computing/met-computing layers cP1, cP2, mcP2, and the SOO-Platform with meta-computing layers mcP-1, mcP1-2, mcP1-3, mcP2, and mcP3; and the hybrid of the previous three—intergrids (iGrids). Note that the MM-Platform is a virtual federation of processors (roughly CPUs) that execute submitted executable codes with the help of a resource broker. Either an SPO-Platform or SOO-Platform federation of services is managed by the form of middleware (operating system), however in the SPO case the application servers are used, but in the SOO case the federations of dynamic autonomous service objects are managed by the relevant operating system. Thus, the latter approach requires a service object-oriented methodology while in the former case the conventional client/server programming is sufficient. The hybrid of three platform abstractions allows for an iGrid to execute both programs and meta-programs as depicted in Fig. 4.4, where platform layers $P1$, $P2$, and $P3$ correspond to resources, resource management, and programming environment correspondingly.

One of the first SOO-Platform was developed under the sponsorship of the National Institute for Standards and Technology (NIST)—the FIPER [26, 31]. The goal of FIPER is to form a federation of distributed service objects that provide engineering data, applications, and tools in a network. A highly flexible software architecture had been developed (1999–2003), in which engineering tools like computer-aided design (CAD), computer-aided engineering (CAE), product data management (PDM), optimization, cost modeling, etc., act as both federating service providers and service requestors [32–37].

SORCER builds on top of FIPER to introduce a meta-computing operating system with all system services necessary, including service management (rendezvous services), a federated file system, and autonomic resource management, to support service-object oriented meta-programming. It provides an integrated solution for complex meta-computing applications. The SORCER meta-computing environment adds an entirely new layer of abstraction to the practice of meta-computing—exertion-oriented (EO) mogramming with a FMI. The EO mogramming makes a positive difference in service-oriented programming primarily through a new federated programming and modeling abstractions as experienced in many service-oriented computing projects including systems deployed at GE Global Research Center, GE Aviation, AFRL, SORCER Lab, and SORCER partners in China and Russia.

4.4.2 Service-Object Oriented Platform: SORCER

The SORCER is a federated service-to-service (S2S) meta-computing environment that treats service providers as network peers with well-defined semantics of a federated SOOA that is based on the FMI. It incorporates Jini semantics of services [16] in the network and the Jini programming model with explicit leases, distributed events, transactions, and discovery/join protocols [16]. While Jini focuses on service management in a networked environment, SORCER is focused on EO mogramming and the execution environment for exertions (see the service platform in Fig. 4.3 as the meta-model of the architecture presented in Fig. 4.5). The abstract model depicted in Fig. 4.3 is the unifying representation for three concrete programming models: imperative languages for IS platforms (executable codes), the SORCER Java API for object platform [27], Exertion-Oriented Language (EOL)

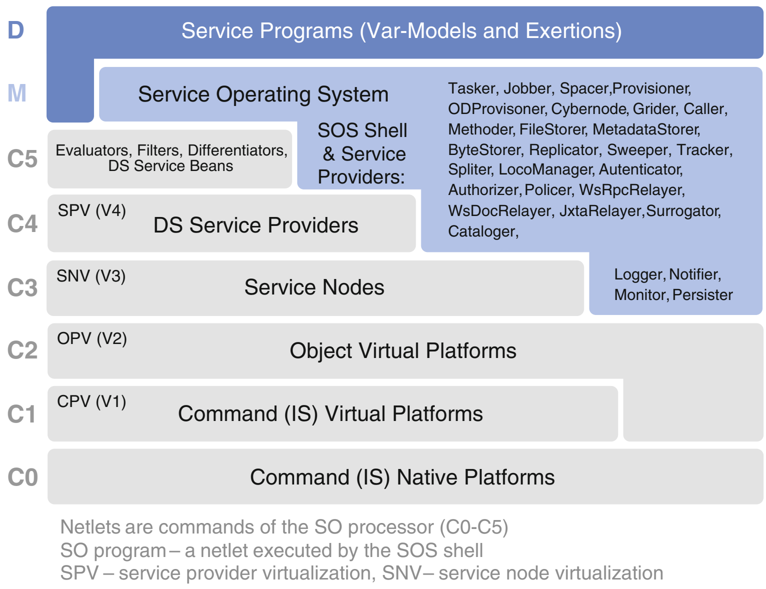


Fig. 4.5 The SORCER layered architecture, where *C*0–*C*5 (carrier)—the metaprocessor with its service cloud at *C*4 and *C*3, platform cloud at *C*2 and *C*1, *M* (management)—SOS, *D* (domain)—service requestors; where *PV* and *OV* stands for provider and object virtualization respectively with the prefix *S* for service, *O* for object, and *C* for command

and Var-oriented Modeling Language (VML) for the SOS [25]. The notation of functional composition has been developed for both EOL and VML, which are usually complemented with the Java object-oriented syntax. Modeling in SORCER is emphasized in Fig. 4.3 by the top layer of DMC triplets: par, par-model, var, and var-model [24]. More details on EOL and VML can be found in source [25].

EOP is a service-oriented programming paradigm using service providers and service commands. Service commands—exertions—are interpreted by the SOS (M layer in Fig. 4.5) and represented by hierarchical data structures that consist of a data context, multiple service signatures, and a control context—to design distributed applications and computer programs. In EOP a service invocation on a provider is determined by a *service signature*. The signature usually includes the service type, operation of the service type, and expected quality of service (QoS). While exertion's signatures identify (match) the required collaborating providers, the control context defines for the SOS how and when the signature operations are applied to the data context.

An exertion is an expression of a distributed process that specifies for the SOS how service collaboration is actualized by a collection of providers playing specific roles used in a specific way [38]. The collaboration specifies a collection of cooperating providers—the exertion federation—identified by the exertion's signatures. Exertions encapsulate explicitly data, operations, and control strategy for the collaboration. The signatures are dynamically bound to corresponding service providers—members of the exerted collaboration.

The exerted members in the federation collaborate transparently according to their control strategy managed by the SOS. The SOS invocation model is based on the Triple Command Pattern [19] that defines the FMI.

Var-Oriented Programming (VOP) is a programming paradigm using service variables called *vars*—data structures defined by the triplet (fidelity) <evaluator, getter, setter> together with a var composition of evaluator's dependent variables—to design service-oriented programs and models. It is based on dataflow principles that changing the value of a var should automatically force recalculation of the values of vars that depend on its value. VOP promotes values defined by evaluators/getters/setters to become the main concept behind any processing. Getters play the role of filters and setters of vars persist values of their vars. Each var might have multiple fidelities selected dynamically during modeling/simulation analyses.

VOM is a modeling paradigm using vars in a specific way to define heterogeneous service federations of var-oriented models, in particular large-scale multidisciplinary analysis (MDA) models including response, parametric, and optimization models [39, 40]. The programming style of VOM is declarative. Models describe the desired results of the program without explicitly listing command or steps that need to be carried out to achieve the results. VOM focuses on how vars connect, unlike imperative programming, which focuses on how evaluators calculate. VOM represents models as a series of interdependent var connections, with the evaluators/getters/setters between these connections being of secondary importance.

The SORCER service requestors (D layer in Fig. 4.5) are expressed in three concrete programming syntaxes: the SORCER Java API [27], the functional composition form (EOL and VML) [25], and the graphical form [41]. The convergence of VML, and EOL into a service-oriented process expression is described in details in [25].

An Exertion is actualized by calling its exert operation. The SORCER FMI defines the following three related operations:

1. `Exertion#exert(Transaction):Exertion`—join the federation; the activated exertion binds to the available provider specified by the exertion's PROCESS signature;
2. `Servicer#service(Exertion, Transaction):Exertion`—requesting the service federation initiated at runtime by the bounding provider from (1) above; and
3. `Exerter#exert(Exertion, Transaction):Exertion`—invoked by all providers in the federation from (2) for their own component exertions. Each component exertion of the parent exertion from (1) is processed as in (2) above.

The above Triple Command Pattern [19] defines three key SORCER interfaces: `Exertion` (metaprogram), `Service` (S2S provider), and `Exerter` (domain-specific service provider specified by the exertion signature).

This approach allows for the S2S environment [38] via the `Service` interface, extensive modularization of Exertions and Exerters, and extensibility from the triple design pattern so requestors can submit onto the network any EO program they want with or without transactional semantics. The Triple Command pattern is used as follows:

1. An exertion is actualized by calling `Exertion#exert()`. The exert operation implemented in `ServiceExertion` uses `ServiceAccessor` to locate in runtime the provider matching the exertion's PROCESS signature.
2. If the matching provider is found, then on its access proxy the `Service#service(Exertion)` method is invoked.
3. When the requestor is authenticated and authorized by the provider to invoke the method defined by the exertion's PROCESS signature, then the provider calls its own exert operation: `Exerter#exert(Exertion)`.
4. `Exerter#exert(Exertion)` operation is implemented by `ServiceTasker`, `ServiceJobber`, `ServiceSpacer`, and `ServiceConcatenator`. The `ServiceTasker` calls by the reflection the domain-specific operation given in the PROCESS signature of its argument exertion. All operations of provider's service type have a single argument: a `Context` type parameter and a `Context` type return value. Each service type is a Java interface implemented by the domain-specific service provider [15]. Tasker, Concatenator, Jobber, and Spacer are rendezvous system services of the SOS. They manage dynamic service federations (batch, block, workflow) of service providers in the network [24].

4.4.3 SORCER Grid Platform: SGrid

To use legacy applications, SORCER supports a traditional approach to grid computing similar to those found in Condor [29] and Globus [18]. The SORCER-based incarnation is called SORCER grid or in short SGrid. Here, instead of exertions being executed by services providing business logic for collaborating exertions, the business logic comes from the service requestor's executable codes that seek compute resources in the network.

The SGrid services in the SORCER environment include Griders accepting exertions and collaborating with Jobbers, Spacers, and Concatenators as SGrid schedulers. Caller and Methoder service providers are used for task execution received from Concatenators, Jobbers, or pulled up from exertion space via Spacers. Callers execute provided codes via a system call as described by the standardized Caller's service context of the submitted task. Methoders download required Java code (task method) from requestors to process a submitted data context with the downloadable code specified in the requestor's exertion signature. In either case, the business logic comes from requestors; it is executable code specified directly or indirectly in the service context used by Callers, or mobile Java code executed by Methoders that is annotated by exertion signatures.

The SGrid with Methoders was used to deploy an algorithm called Basic Local Alignment Search Tool (BLAST) [42] to compare newly discovered, unknown DNA and protein sequences against a large database with more than three gigabytes of known sequences [43]. BLAST (C++ code) searches the database for sequences that are identical or similar to the unknown sequence. This process enables scientists to make inferences about the function of the unknown sequence based on what is understood about the similar sequences found in the database. Many projects at the USDA–ARS Research Unit, for example, involve as many as 10,000 unknown sequences, each of which must be analyzed via the BLAST algorithm. A project involving 10,000 unknown sequences requires about three weeks to complete on a single desktop computer. The S-BLAST implemented in SORCER [43], a federated form of the BLAST algorithm, reduces the amount of time required to perform searches for large sets of unknown sequences to less than one day. S-BLAST is comprised of BlastProvider (with the attached BLAST Service UI), Jobbers, Spacers, and Methoders. Methoders in S-BLAST download Java code (to execute a task operation) that initializes a required database before making a system call on the BLAST code. Armed with the S-BLAST's SGrid and seventeen commodity computers, projects that previously took three weeks to complete can now be finished in less than one day.

4.4.4 SORCER Intergrid Platform: IGrid

In Sects. 4.4.2 and 4.4.3 two complementary platforms: SORCER and SGrid are described respectively. As shown in Fig. 4.6 a hybrid of three types of platforms is feasible to create intergrid (iGrid) applications that take advantage of both SORCER and SGrid platforms, and SPO-Platform synergistically. Legacy applications can be reused directly in SGrid or any MM-Platform and new complex, for example concurrent engineering, applications [44, 45] can be defined in SORCER.

Relayers are SORCER gateway providers that transform exertions to native representations and vice versa. The following exertion gateways have been developed: JxtaRelayer for JXTA and WsRpcRelayer and WsDocRelayer for for RPC and document style Web services, respectively. Relayers exhibit native and SORCER Grid behavior by implementing dual protocols. For example a JxtaRelayer (1) in Fig. 4.6 is at the same time a `Service` (1-) and a JXTA peer (1-) implementing JXTA interfaces. Therefore it shows up also in SORCER Grid and in the JXTA Grid as well. Native Grid providers can play the SORCER role (as SORCER wrappers), thus are available in the iGrid along with SORCER providers. For example, a JXTA peer 4- implements the `Service` interface, so shows up in the JXTA iGrid as provider 4. Also, native Grid providers via corresponding relayers can access iGrid services (bottom-up) Thus, the iGrid is a projection of Services onto meta-compute and compute Grids.

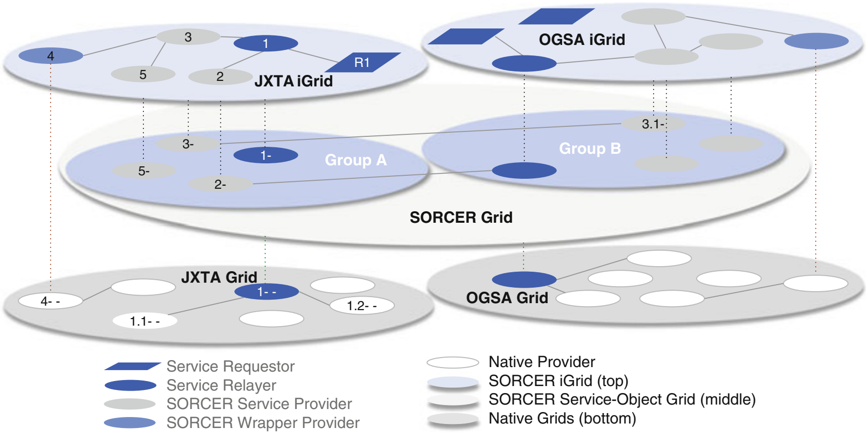


Fig. 4.6 Integrating and wrapping native Grids with SORCER Grids (Group A and Group B). Two requestors, one in JXTA iGrid, one in OGSA iGrid submits exertion to a corresponding relayer. Two federations are formed that include providers from the two horizontal layers below the iGrid layer (as indicated by *continuous* and *dashed links*)

The iGrid-integrating model is illustrated in Fig. 4.6, where horizontal native technology grids (bottom) are seamlessly integrated with horizontal SORCER metacopute Grids (top) via the SOS services in Sorcer Grid (middle). Through the use of open standards-based communication—Jini, Web Services, Globus/OGSA, and Java interoperability—iGrid leverages the federated SOOA with its inherent neutrality of provider's protocol, location, and implementation along with the flexibility of EO mogramming and meta-compute federated OS.

4.5 A Case Study of an Efficient Supersonic Air Vehicle

The AFRL's Multidisciplinary Science and Technology Center (MSTC) is investigating conceptual design processes and computing frameworks that could significantly impact the design of the next generation of efficient supersonic air vehicle (ESAV). To make the technological advancements required of a new ESAV, the conceptual design process must accommodate both low- and high-fidelity trans-disciplinary engineering analyses. These analyses may be coupled and computationally expensive, which poses a challenge since a large number of configurations must be analyzed. In light of these observations, the ESAV design process was implemented using the SOS to combine propulsion, structures, aerodynamics, performance, and aero-elasticity in a MDA [40, 46]. The SORCER platform provides MDA automation and flexible service-oriented integration to distributed computing resources necessary to achieve the volume of analyses required for conceptual design.

While most service systems are based on the SPOA the SORCER platform utilizes the SOOA architecture. That makes SORCER exhibiting the following unique features with respect to existing grid and cloud systems:

1. Smart proxying for balancing business logic execution between a service requestor and provider with fat proxying for running the provider's code completely at the requestor side;
2. A self-healing runtime environment using network discover/join protocols for dynamic lookup services;
3. Location/implementation neutrality, but most importantly wire protocol neutrality and transport protocol selection at service deployment (transport endpoints);
4. The front-end mogramming with the capability of both back and frontend service provider development;
5. Unification of SO procedural (EOP) with SO functional composition (par-oriented and var-oriented-modeling);
6. Ease of parallelization with self-balancing exertion space computing and transactional semantics;
7. Front-end choice of PUSH or PULL execution of nested exertions;
8. Front-end (on-demand) autonomic service provider provisioning/unprovisioning;
9. Context awareness of the service-oriented computing with interoperability across service federations; and
10. Code mobility across service federations—dynamic behavioral transfer between requestors and providers.

The MDA is a blend of conceptual and preliminary design methods from propulsion, structures, aerodynamics, performance, and aero-elasticity disciplines. The process begins by parametrically generating discretized geometry suitable for several different analyses at varying fidelities. The geometry is used as input to compute several figures of merit of the aircraft, which include the aircraft drag polars, design mass, range, and aero-elastic performance. The different responses are evaluated for several flight conditions and maneuvers. These responses are then used to construct the objective and constraints of the multidisciplinary optimization (MDO) problem.

MDO generally requires a large number of MDAs be performed. This significant computational burden is addressed by using the SORCER platform. The SO and network-centric approach of SORCER enables the use of heterogeneous computing resources, including a variety of operating systems, hardware, and software. Specifically, the ESAV studies performed herein use SORCER in conjunction with a mix of Linux-based cluster computers, desktop Linux-based PCs, Windows PCs, and Macintosh PCs. The ability of SORCER to leverage these resources is significant to MDO applications in two ways: (1) it supports platform-specific executable codes that may be required by an MDA; and (2) it enables a variety of computing resources to be used as one entity (including stand-alone PCs, computing clusters, and high-performance computing facilities). The main requirements for using a computational resource in SORCER are network connectivity and Java platform compatibility. SORCER also supports load balancing across computational resources using space computing, making the evaluation of MDO objective and constraint functions in parallel a simple and a dynamically scalable process.

SOS Spacer providers enable different processes on different computers to communicate asynchronously with transactional semantics in a reliable manner [38]. Using Spacer services, SOS implements a self-load balancing app cloud (see Fig. 4.4) that can dynamically grow and shrink during the course of an optimization study, see an ESAV example Fig. 4.6-left. Various service providers or multiple instances of the same multifunctional (implementing multiple services types) service provider can be configured for parallelization in deployment/provisioning accordingly to compute power of their hosting environment (laptop, workstation, cluster, and supercomputer).

An exertion space or “space” is exertion storage in the network that is managed by the SOS. The space provides a type of shared memory where requestors can put exertions they wish to be processed by service providers. Service providers, in turn, continuously lookup up exertions in their space (also SOS service) to be processed. If a service provider sees a task it can operate on in its space, and the task has not been processed yet, the provider takes the task (removes it) from the space. The provider then performs the requested service and returns the task to the space as completed. Once the task has been returned to the space, SOS Spacer that initially wrote the task to the space detects the completed task then takes the task from the space and returns it to the submitting service requestor. Pars and vars are frequent SORCER requestors with their invokers and evaluators as space exertions. This way par-oriented and var-oriented models access various applications, tools, and utilities as ubiquities dynamic services in the network as illustrated in Fig. 4.7.

The ESAV service providers are used with an external optimization program as the SORCER requestor (Matlab Client, Fig. 4.7) to optimize an ESAV for range. The optimized design has a higher aspect ratio than the baseline design. The received results provide a degree of validation of the optimization code implementation, the SORCER ESAV parametric model, the SORCER providers, and the SOS [46].

The use of the space computing proved reliable and efficient. It was a straightforward process to add computers to the SORCER service Cloud as needed during the course of the two optimization studies. This flexibility proved valuable as the number of computers available varied from day-to-day. Parallelization of var-responses in parametric models with exertion space coordination resulted in significant savings. In this case it reduced the computational time to perform the optimization from 24 h to proximately 2 h [47]. The reduction is achieved mainly by the parallelization of SORCER parametric models for each parametric response (a vector of var values) and parallelization of var exertion evaluators (for each vector) executed using exertion space as illustrated in Fig. 4.7.

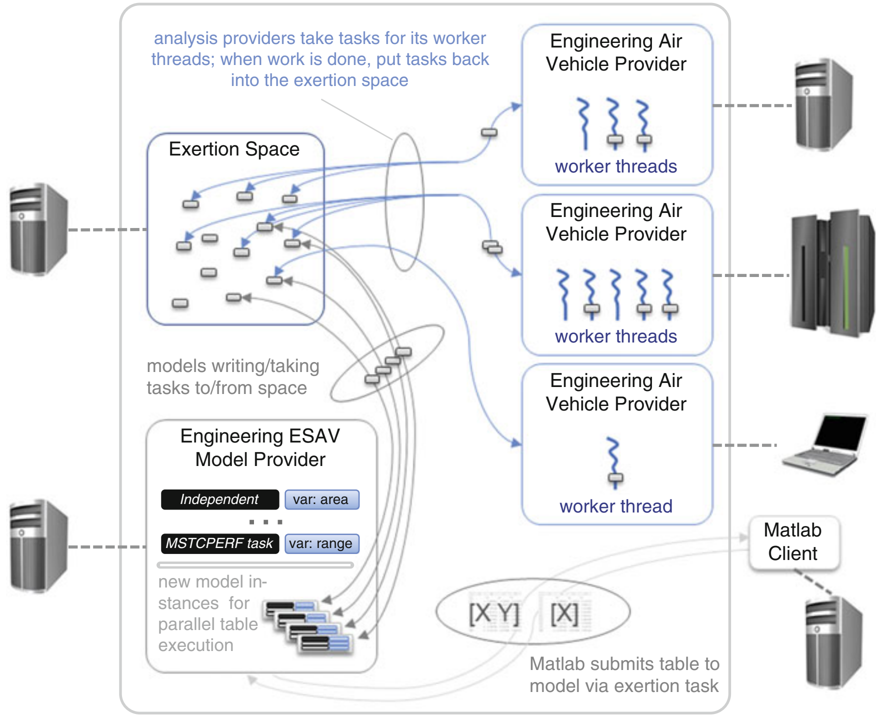


Fig. 4.7 SORCER provides exertion space for a flexible, dynamic space computing for ESAV optimization studies. Exertions of variables in the parametric model are written into the space when variable values are needed. The exertions from the space are read and processed by engineering air vehicle service providers that return the results into the apace to be collected by the model for the requestor's optimization program

4.6 Conclusions and Outlook

To work effectively in large, distributed environments, concurrent engineering teams need a service-oriented programming methodology along with the common design process, domain-independent representations of designs, and general criteria for decision making. Distributed MDA and optimization are essential for decision making in engineering design that provide a foundation for service oriented concurrent engineering [44, 45].

As we move from the problems of the information era to more complex problems of the molecular era, it is becoming evident that new programming languages are required. These languages should reflect the complexity of meta-computing problems we are facing already in the molecular era, for example, concurrent engineering processes of collaborative design by hundreds or thousands of people working together and using thousands of programs written already in software

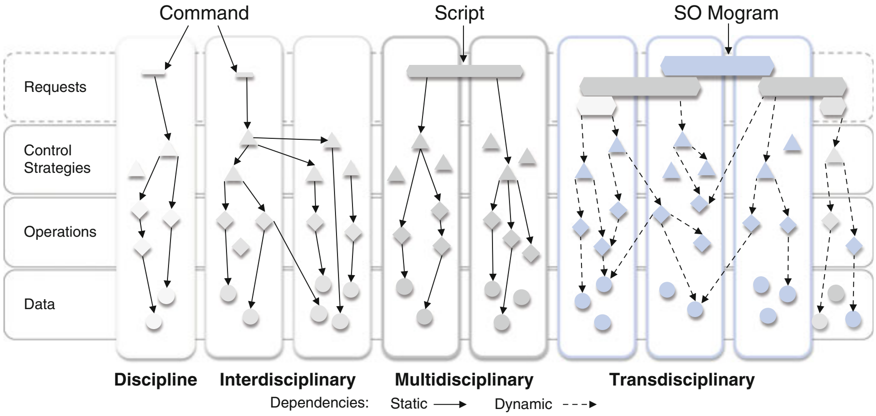


Fig. 4.8 By providing easy-to-use, self-discovering services representing domain knowledge (data), tools (operations), and related technologies (control) with exertion-oriented and var-modeling (mogramming) methodology, the SORCER environment reduces integration and deployment costs, facilitate productivity, increases research collaboration, and advances the development and acceptance of secure and fault tolerant cross-disciplinary concurrent engineering solutions

languages that are located around the globe. The cross-disciplinary design of an aircraft engine or even a whole air vehicle requires dynamic large-scale cross-disciplinary meta-computing systems (see Fig. 4.8).

The EOP introduces the new abstraction of service-oriented programming with service providers and exertions instead of object-oriented conventional objects and messages. An exertion not only encapsulates signatures, data context, and control strategy, it encapsulates the matching federation of service providers as well. From the meta-computing platform point of view, exertions are entities considered at the programming level, service interactions at the operating system level, and federations at the processor level. Thus, exertions are process expressions that define service collaborations. The SOS manages the collaborations as FMI interactions on its virtual processor—the dynamically formed service federations.

Service providers can be easily deployed in SORCER by injecting implementation (executable code) of domain-specific interfaces into the FMI framework. The providers register proxies, including smart proxies, via dependency injection defined during their deployment. Executing the exertion, by sending it onto the network, means forming a required federation from currently available or provisioned service providers at runtime. Service providers in the federation work on service contexts of all component exertions collaboratively as specified by the control strategies of the corresponding component exertions.

The FMI framework defines federated SOA and allows for the P2P computing via the `Service` interface, extensive modularization of Exertions and Exerters, and extensibility from the Triple Command design pattern. The presented EOP methodology with the SOS, its federated file system (SILENUS/FICUS), and resource management framework (SERVME) has been successfully deployed and tested in multiple concurrent engineering and large-scale distributed applications [31, 32, 34–37, 46–51].

The presented description of the exertion-oriented methodology can be finalized as follows:

1. Thinking more explicitly about programming languages (DSL languages for humans as VML and EOL) instead of software languages (languages for computers) may be our best tool for dealing with real world complexity.
2. Understanding the principles that run across process expressions and appreciating which language features and related computing platforms are best suited for which type of process, bring these process expressions to useful life, e.g., seamless federations of tools, applications, and utilities in concurrent engineering processes.
3. No matter how complex and polished the individual process operations are, it is often the quality of the operating system that determines the power of the computing system. Note that the SOS is the service-oriented operating systems for exertion-oriented mogramming or generic middleware for SGrid and iGrid.
4. It provides unified programming and modeling environment. Procedural service-oriented programming enables bottom up problem solving and VOM enables top-down problem solving [39, 40] depicted in Fig. 4.9.

The presented description suggests that mixing both a process expression and implementation components (service providers) within a single computing platform and with the same programming language for both introduces inefficiencies and complexity of large transdisciplinary computing systems beyond human

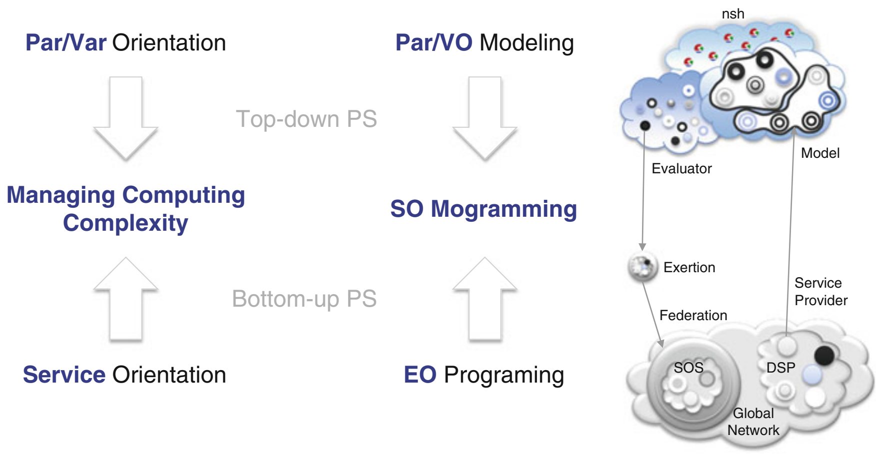


Fig. 4.9 Managing transdisciplinary complexity with convergence of *top-down* service-oriented modeling and *bottom-up* service-oriented programming (*right* exertions in models as exertion evaluators and models as service providers in exertions)

comprehension. The proposed solution is to use the DCM and MCM architectures for implementation of transdisciplinary systems with the service-object oriented platform for coherent management of various heterogeneous component computing platforms.

Complex adaptive designs involve hundreds of low-level designs and simulations with thousands of programs written already in software languages (languages to create executable codes that are dislocated around the globe and have to be integrated into meta-applications written in DSL expressing problem to be solved by humans for human beings.

DSLs are for humans, intended to express specific complex problems, related processes, and corresponding solutions. In SORCER two basic programming languages for transdisciplinary computing are EOL and VML. These languages are interpreted by the SOS shell as service-oriented commands. The concept of the evaluator/getter/setter triplet in modeling provides the uniform service-orientation for all computing and meta-computing needs with various engineering applications, tools, and utilities.

The SORCER platform with three layers of converged programming: exertion-oriented (for service collaborations), and VOM (multidisciplinary var-oriented models with multi-fidelity compositions) has been successfully deployed and tested for the SO space exploration and parametric and optimization mogramming in recent application at AFRL/WPAFB [46, 47, 49, 51–53].

Acknowledgments This work was partially supported by Air Force Research Lab, Aerospace Systems Directorate, Multidisciplinary Science and Technology Center, the contract number F33615-03-D-3307, Algorithms for Federated High Fidelity Engineering Design Optimization, Polish-Japanese Institute of IT (the contract number POIG.01.04.00-14-062/12 the National Center for Research and Development), and the National Natural Science Foundation of China (Project No. 51175033).

References

1. Dwivedi SN, Sobolewski M (1990) Concurrent engineering—an introduction. In: Proceedings of the fifth international conference on CAD/CAM robotics and factories of the future '90. Concurrent engineering, vol 1. Springer, New York, pp 3–16
2. Markov AA (1971) Theory of algorithms (trans by Schorr-Kon JJ). Keter Press, Israel
3. Kleppe A (2009) Software language engineering: creating domain-specific languages using metamodels. Pearson Education, Boston
4. N. N (1988) The role of concurrent engineering in weapon systems acquisition. IDA Report R-338, Dec 1988
5. Sobolewski M (1990) In: Percept knowledge and concurrency. Proceedings of the second national symposium on concurrent engineering, West Virginia University, Morgantown, pp 111–137
6. Postel J, Sunshine C, Cohen D (1981) The ARPA internet protocol. Comput Netw 5:261–271
7. Lynch D, Rose MT (eds) (1992) Internet system handbook. Addison-Wesley, Reading
8. Postel J, Reynolds J (1987) Request for comments reference guide (RFC1000). Internet Engineering Task Force

9. Lee J (ed.) (1992) Time-sharing and interactive computing at MIT. IEEE Ann Hist Comput 14:1
10. Hafner K, Lyon M (1996) Where wizards stay up late. Simon and Schuster, New York
11. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid: enabling scalable virtual organizations. Int J Supercomputer Appl 15(3):200–222
12. Linthicum DS (2009) Cloud computing and SOA convergence in your enterprise: a step-by-step guide. Addison-Wesley Professional, Boston
13. Birrell AD, Nelson BJ (1983) Implementing remote procedure calls. XEROX CSL-83-7
14. Ruh WA, Herron T, Klinker P (1999) IIOP complete: understanding CORBA and middleware interoperability. Addison-Wesley, Boston
15. Pitt E, McNiff K (2001) java.rmi: the remote method invocation guide. Addison-Wesley Professional, Boston
16. Newmarch J (2006) Foundations of Jini 2 programming. Apress, Berkeley, ISBN-13: 978-1590597163
17. McGovern J, Tyagi S, Stevens ME, Mathew S (2003) Java web services architecture. Morgan Kaufmann, San Francisco
18. Sotomayor B, Childers L (2005) Globus® toolkit 4: programming java services. Morgan Kaufmann, San Francisco
19. Sobolewski M (2007) In: Federated method invocation with exertions. Proceedings of the IMCSIT conference. PTI Press, ISSN 1896-7094, pp 765–778
20. Sobolewski M (2008) SORCER: computing and metacomputing intergrid. In: 10th international conference on enterprise information systems, Barcelona, Spain. Available at: http://sorcer.cs.ttu.edu/publications/papers/2008/C3_344_Sobolewski.pdf
21. Sobolewski M (2009) Metacomputing with federated method invocation. In: Hussain MA (ed) Advances in computer science and IT. In-Tech, Rijeka, pp 337–363. http://www.intechopen.com/books/advances-in-computer-science-and-it/metacomputing-with-federated-method-invocation. Accessed 15 Feb 2014
22. Sobolewski M (2010) Exerted enterprise computing: from protocol-oriented networking to exertion-oriented networking. In: Meersman R et al (eds) OTM 2010 workshops, LNCS 6428, 2010. Springer, Berlin, pp 182–201
23. Sobolewski M (2011) Provisioning object-oriented service clouds for exertion-oriented programming. In: The 1st international conference on cloud computing and services science, CLOSER 2011, Noordwijkerhout, The Netherlands, 7–9 May 2011, SciTePress Digital Library. http://sorcersoft.org/publications/papers/2011/CLOSER_2011_KS.pdf. Accessed 15 Feb 2014
24. Sobolewski M (2014) Service oriented computing platform: an architectural case study. In: Ramanathan R, Raja K (eds) Handbook of research on architectural trends in service-driven computing, IGI Global, Hershey
25. Sobolewski M, Kolonay RM (2012) Unified mogramming with var-oriented modeling and exertion-oriented programming languages. Int J Commun Netw Syst Sci 5:579–592. Published Online Sep 2012 (http://www.SciRP.org/journal/ijcns)
26. Sobolewski M (2002) Federated P2P services in CE environments, advances in concurrent engineering. A.A. Balkema Publishers, Boca Raton, pp 13–22
27. Sobolewski M (2008) Exertion oriented programming. Int J Comput Sci Inf Syst 3(1):86–109
28. Sobolewski M (2010) Object-oriented metacomputing with exertions. In: Gunasekaran A, Sandhu M (eds) Handbook on business information systems. World Scientific, Singapore
29. Thain D, Tannenbaum T, Livny M (2003) Condor and the grid. In: Berman F, Hey AJG, Fox G (eds) Grid computing: making the global infrastructure a reality. Wiley, Chichester
30. Juric MB, Benny M, Sarang P (2006) Business process execution language for web services BPEL and BPEL4WS, 2nd edn. Packt Publishing, Birmingham
31. Röhl PJ, Kolonay RM, Irani RK, Sobolewski M, Kao K (2000) A federated intelligent product environment. In: AIAA-2000-4902, 8th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, Long Beach

32. Kolonay RM, Sobolewski M, Tappeta R, Paradis M, Burton S (2002) Network-centric MAO environment. In: The society for modeling and simulation international, western multiconference, San Antonio
33. Sampath R, Kolonay RM, Kuhne CM (2002) 2D/3D CFD design optimization using the federated intelligent product environment (FIPER) technology. In: AIAA-2002-5479, 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Atlanta, GA
34. Kao KJ, Seeley CE, Yin S, Kolonay RM, Rus T, Paradis MJ (2003) Business-to-business virtual collaboration of aircraft engine combustor design. In: Proceedings of DETC'03 ASME 2003 design engineering technical conferences and computers and information in engineering conference, Chicago
35. Goel S, Talya S, Sobolewski M (2005) Preliminary design using distributed service-based computing. In: Sobolewski M, Ghodous P (eds) Next generation concurrent engineering. Proceeding of the 12th conference on concurrent engineering: research and applications. ISPE Inc./Omnipress, New York, pp 113–120
36. Goel S, Shashishekara S, Talya S, Sobolewski M (2007) Service-based P2P overlay network for collaborative problem solving. Decis Support Syst 43(2):547–568
37. Goel S, Talya S, Sobolewski M (2008) Mapping engineering design processes onto a service-grid: turbine design optimization. Int J Concurrent Eng Res Appl Concurrent Eng 16:139–147
38. Sobolewski M (2008) Federated collaborations with exertions. In: 17th IEEE international workshop on enabling technologies: infrastructures for collaborative enterprises (WETICE), pp 127–132
39. Sobolewski M, Kolonay R (2013) Service-oriented programming for design space exploration. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, proceedings of the 19th ISPE international conference on concurrent engineering, Springer-Verlag, London, pp 995–1007
40. Sobolewski M, Burton S, Kolonay R (2013) Parametric mogramming with var-oriented modeling and exertion-oriented programming languages. In: Bil C et al (eds) Proceedings of the 20th ISPE international conference on concurrent engineering, IOS Press, pp 381–390, http://ebooks.iospress.nl/publication/34826. Accessed on 9 March 2014
41. Sobolewski M, Kolonay R (2006) Federated grid computing with interactive service-oriented programming. Int J Concurrent Eng Res Appl 14(1):55–66
42. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410
43. Khurana V, Berger M, Sobolewski M (2005) A federated grid environment with replication services. Next Gener Concurrent Eng. In: Sobolewski M, Ghodus P (eds) Next generation concurrent engineering: smart and concurrent integration of product data, services, and control strategies. ISPE/Omnipress, pp 93–103
44. Sobolewski M, Cha J (eds) (2004) Concurrent engineering: the worldwide engineering grid. Tsinghua Press and Springer-Verlag, London
45. Sobolewski M, Ghodous P (eds) (2005) Next generation concurrent engineering: smart and concurrent integration of product data, services, and control strategies. ISPE/Omnipress
46. Burton SA, Alyanak EJ, Kolonay RM (2012) Efficient supersonic air vehicle analysis and optimization implementation using SORCER. In: 12th AIAA aviation technology, integration, and operations (ATIO) conference and 14th AIAA/ISSM AIAA 2012-5520, Indianapolis, Indiana (AIAA 2012-5520), 17–19 Sep 2012
47. Kolonay RM (2013) Physics-based distributed collaborative design for aerospace vehicle development and technology assessment. In: Bil C et al (eds) Proceedings of the 20th ISPE international conference on concurrent engineering. IOS Press, pp 198–215. http://ebooks.iospress.nl/publication/34808. Accessed 15 March 2014
48. Burton SA, Tappeta R, Kolonay RM, Padmanabhan D (2002) Turbine blade reliability-based optimization using variable-complexity method. In: 43rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, Denver, Colorado. AIAA 2002-1710

49. Kolonay RM, Thompson ED, Camberos JA, Eastep F (2007) Active control of transpiration boundary conditions for drag minimization with an euler CFD solver. In: AIAA-2007-1891, 48th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, Honolulu, Hawaii
50. Xu W, Cha J, Sobolewski M (2008) A service-oriented collaborative design platform for concurrent engineering. Adv Mater Res 44–46(2008):717–724
51. Kolonay RM, Sobolewski M (2011) Service ORiented Computing EnviRonment (SORCER) for large scale, distributed, dynamic fidelity aeroelastic analysis & optimization, international forum on aeroelasticity and structural dynamics. In: IFASD 2011, Paris, 26–30 June 2011
52. Kolonay RM (2014) A physics-based distributed collaborative design process for military aerospace vehicle development and technology assessment. Int J Agile Syst Manage 7(3/4): 242–260
53. Sobolewski M (2014) Unifying front-end and back-end federated services for integrated product development. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering, IOS Press, Amsterdam, pp 3–16

Part II
New Developments and Methods

Chapter 5
Requirements Engineering

Stefan Wiesner, Margherita Peruzzini, Jannicke Baalsrud Hauge and Klaus-Dieter Thoben

Abstract Requirements engineering (RE) is the key to success or failure of every product, service or system development project, understanding the development results as the implementation of the specific set of requirements. A good requirements definition is thus the prerequisite for high-quality solutions and reduces the cost of change, both of prototypes and production tools, and ultimately the warranty costs. However, RE for system development is more and more challenged by two interrelated trends: the increasing complexity of systems and the responsibility of the provider for the whole system life cycle. Thus, from a systems engineering point of view, RE has to define requirements for a rising amount of tangible and intangible components from a growing number of different stakeholders. Additionally, RE has to take into account requirements from every stage of the system life cycle and feed the results back to the development process. Many organizations are still missing effective practices and a documented RE process to tackle the upcoming challenges in systems engineering. This chapter aims at giving an overview on the RE context and challenges for systems engineering and subsequently describes the state-of-the-art for structuring and processing requirements. Furthermore, two case studies illustrate the current situation and methods for resolution in industry and

S. Wiesner (✉) · J.B. Hauge · K.-D. Thoben
BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Hochschulring 20, 28359 Bremen, Germany
e-mail: wie@biba.uni-bremen.de

J.B. Hauge
e-mail: baa@biba.uni-bremen.de

K.-D. Thoben
e-mail: tho@biba.uni-bremen.de

M. Peruzzini
Indesit Company S.p.A, Viale A. Merloni 47, 60044 Fabriano AN, Italy
e-mail: margherita.peruzzini@consultants.indesit.com; m.peruzzini@univpm.it

M. Peruzzini
Department of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, via Brecce Bianche 12, 60131 Ancona, Italy

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_5

show how the identified challenges can be met by IT support. Finally, future trends and needs for RE research and its further integration with concurrent engineering and life cycle management approaches are outlined.

Keywords Requirements engineering · Systems engineering · Life cycle management

5.1 Introduction

Nowadays, the field of systems engineering is influenced by rapid technological change and ever growing competition. In order to be the first to react on market trends, methodologies like concurrent engineering (CE) have been introduced to shorten development cycles and reduce "time to market" [1, 2]. However, just being fast is not sufficient; additionally the system has to be both appropriate and cost effective [3]. Customers demand integrated solutions and services, covering the whole system life cycle, from ideation to decommission. The required competencies for system development and support in other life cycle phases are included through collaboration with partners from different domains [4]. This increases both the number of stakeholders involved in systems engineering and the complexity of the system itself. Consequently, understanding what the customer and the other affected stakeholders expect from the system, i.e. their underlying needs, and linking information from all phases of the life cycle to the development process is a prerequisite for successful systems engineering [5–7].

Requirements define the needs of organizations, groups or people along with their surroundings and describe what a solution must offer in order to satisfy those needs. Their formulation, documentation and maintenance are the main objectives of requirements engineering (RE). It describes "*a process, in which the needs of one or many stakeholders and their environment are determined to find the solution for a specific problem*" [8]. Inadequate RE is one of the main sources for the failure of development projects and culminates in exceeding budgets, missing functionalities or even the abortion of the project [9]. However, the importance of RE is often underrated, leading to errors in the requirements specification. Requirements on requirements, like completeness, consistency, verifiability etc., are disregarded. Errors in requirements specification are regularly only discovered in later development phases. The later they are discovered, the higher the costs of correcting the errors [10].

This chapter aims at giving an overview of the current state-of-the-art in RE research and application in industry, as well as providing an outlook on future research perspectives in connection with concurrent systems engineering and life cycle management. Therefore, Sect. 5.2 describes the systems engineering context in which RE is applied and identifies challenges related to the growing complexity of systems and life cycle responsibility. Different levels in systems engineering use different types of requirements, for which Sect. 5.3 presents a generic structure along with detailed definitions. Section 5.4 shows how the different types of requirements are identified, specified and validated during the RE process, while supporting IT tools for RE are discussed in Sect. 5.5. Two case studies illustrating the current application of RE methodologies and tools in different industries are presented in Sect. 5.6. Based on the findings from theory and practice, Sect. 5.7 finally summarizes the results and gives an outlook on future RE perspectives with regards to the identified gaps and challenges.

5.2 Context and Challenges of Requirements Engineering

This section outlines the background of RE as a discipline and highlights its growing importance and interrelationships with systems engineering and the system life cycle. Furthermore, it discusses the current challenges for RE and relates it to other relevant research topics.

Every development project is based on requirements that define what the targeted beneficiaries expect as a result. They are needed for planning the development process, assessing the impact of changes and testing the acceptance of the outcomes [11]. Consequently, the RE process starts at the beginning of each development project. Traditional development approaches that originate from the manufacturing domain often see RE as a discrete development phase with the objective of creating the requirements specification. With the emergence of software development, this sequential approach was adapted as the only available formalized methodology, e.g. as being described in the waterfall model [12]. The requirements elicited for the project are documented in a requirements specification. This document serves as reference for the subsequent development activities, as shown in Fig. 5.1.

However, the traditional view of RE as a discrete phase in the beginning of a development project incorporates a number of substantial disadvantages. If RE is only conducted for a limited period of time, change requests occurring during later development phases are not considered in the requirements specification, leading to an unclear documentation of which parts of the original specification are really implemented at the end. The requirements are engineered for each development project separately. Consequently, they cannot be re-used in other projects, even if the same at the end, which is costly and time consuming [13]. In addition, since the requirements specification might not be congruent with the final implementation, it is difficult to reuse it for change management and testing. This has, among other things, an effect on the development time. First approaches to quantify the stability of requirements based on Function Point Modeling and regression analysis have been developed and applied in practice [14, 15]. Furthermore, focusing only on the current development can lead to ignoring information that is not relevant for this project. However, the information ignored can be important for other development projects. Thus, conducting RE as a discrete phase, like in the waterfall model, is

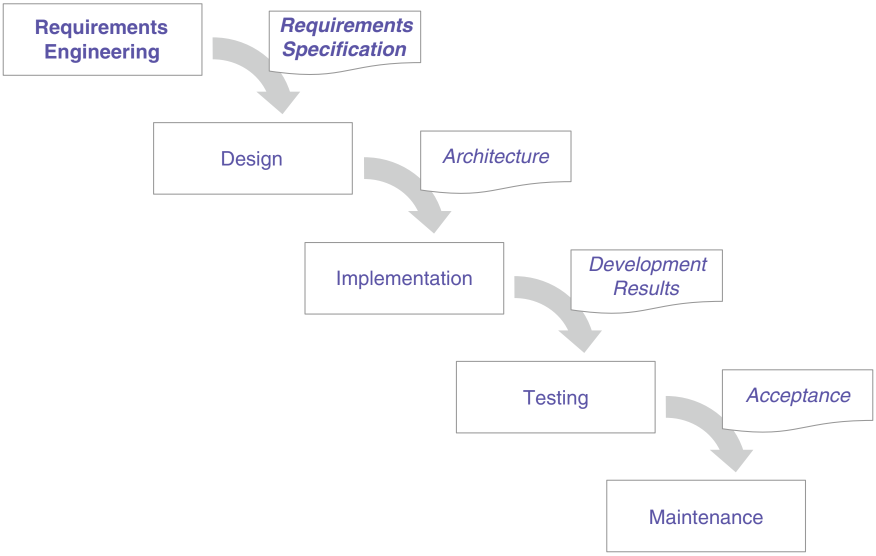


Fig. 5.1 Requirements engineering in the waterfall model

only suitable for stable development projects with unchanging requirements, such as simple products, where risks and problems can be anticipated.

In contrast, systems engineering deals with the development of complex solutions, consisting of a large number of components whose interactions shall produce a desired result [16]. Accordingly, RE for systems is facing a number of additional challenges, which cannot be addressed in a single development phase. In traditional RE scenarios for simple products, the stakeholders are generally aware of their needs. Often, a specific functionality is requested and the product is developed on the basis of formalized requirements by a single enterprise. However, a system is often expected to solve a particular customer problem without prescribing a specific functionality. Cross-linking with other systems and integration into the environment increases the complexity of the system development. In addition, innovative properties of a system are nowadays predominantly realized or even only possible through the integration of software and services. This leads to the situation that stakeholders do not really know what is needed to solve their problem. Sometimes the problem itself cannot even be described in detail [17].

Another challenge is the temporary collaboration of different stakeholders in systems engineering. Besides the customer and user of the system, actors like the project manager, product designers, software developers, service engineers, marketing experts, suppliers, quality assurance and many more have to be involved. This induces a change in RE from a quasi-stable and simple environment to a more complex and dynamic variation. The stakeholder milieu grows in size as well as in complexity, leading to various factors to be considered [18]. Not all stakeholders will be based in the same location; it is possible that some of the stakeholders are globally distributed. This makes collaboration during RE much more difficult, as personal meetings are much harder to arrange. Conflicts and interdependencies have to be assessed for a larger number of requirements. Information exchange between the stakeholders is the key success factor. However, without pre-defined structures, formats and interfaces communication might be chaotic and lead to information loss and delays [19].

Competition also demands for faster implementation of customer wishes and to have innovative solutions more quickly available on the market. Thus, development times have to be reduced in spite of increasing system complexity, while falling prices lead to cost pressure. Demands on system quality and its availability are constantly growing. The system has to fulfill all agreed requirements, notwithstanding shorter development times and high cost pressure. The challenges described above, in combination with the growing amount of requirements for complex systems favour errors in the RE phase, leading to risks for the development process. A weak definition of requirements can slow down system development and induce unnecessary costs for design changes (see Fig. 5.2). If incorrect requirements are identified, an unsuitable system architecture and implementation can result and the system may have missing or wrong functionalities.

The correction of errors in requirements specification discovered during system design can be 20 times more expensive than during RE. Errors discovered during testing can be even 100 times as costly to correct [10]. It is therefore important to discover and correct erroneous requirements as early as possible in the development process.

Consequently, systems engineering has to involve RE more and more as an independent activity not restricted to a specific development phase or project. This would enable a systematic learning process for involved stakeholders as well as the exchange of requirements along the development process and with similar projects. Feedback from the single development projects might constantly be integrated into a knowledge and requirements base that can be used for future development projects as well. Requirements for a new system could be compiled from the currently known requirements, avoiding repeating requirements analyses and thus decreasing

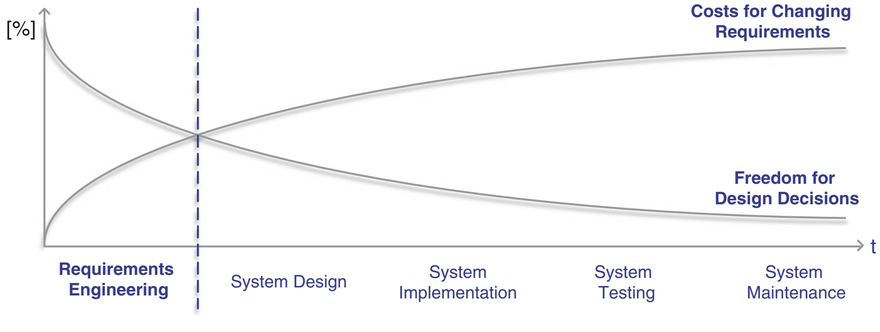


Fig. 5.2 Cost for changing requirements versus freedom for design decisions

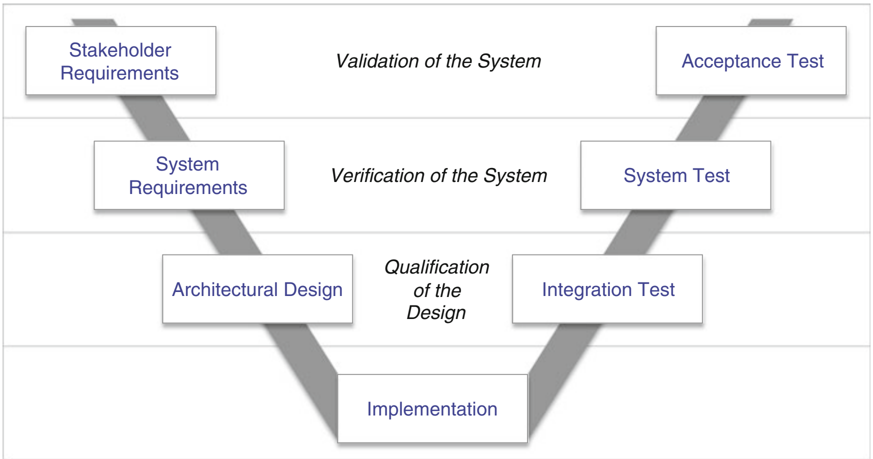


Fig. 5.3 Requirements engineering in the V-Model, according to [11]

the time needed for the development of a system and also improve the quality of the requirement analysis [9]. However, continuous RE requires the establishment of an RE team, being responsible for the development, management and update of the requirements. The team has to have a clear responsibility and all stakeholders require defined contact points for all RE activities. Thus there would be increased need for coordination between the development activities and possibly higher costs for RE [20].

Therefore, in concordance with the principles of CE, RE continues along the development process of a system and secures a consistent and traceable elicitation and management of requirements. There is an ongoing interaction between RE and the development phases in systems engineering, as can be shown with the V-Model in Fig. 5.3.

The term "V-Model" describes a family of models used to illustrate the software or system development process and its main steps [21]. Figure 5.3 shows the activities performed during the individual phases of system development in separate layers. On the left side of the "V", stakeholder requirements are collected and decomposed into system requirements, from which the architectural design of the system is derived. After system implementation, the development results have to be tested against the original specification, which is done during the activities on the right side of the "V". Integration testing qualifies the correctness of the architectural design, while the system test verifies the compliance of the whole system to the technical specification. Finally, an acceptance test validates that the system really meets the needs of the stakeholders [11]. All these activities are related to RE and the following section explains the role of the different types of requirements in the layers of the V-Model in more detail.

5.3 Requirements

In general, *a requirement is a need for a physical attribute or functionality of a solution* [20]. It describes the capabilities or characteristics a product or service has to provide in order to deal with a specific problem. Based on this fundamental description, a prominent definition of a requirement in literature is presented below.

5.3.1 Definition

The Institute of Electrical and Electronics Engineers (IEEE) gives an elaborated definition for requirements [22]:

1. *A condition or capability needed by a user (person or system) to solve a problem or achieve a goal*
2. *A condition or capability, which has to be provided by a system or part of a system, to fulfill a contract, a standard, a specification or any other formal documents*
3. *A documented representation of a condition or capability, as in* 1 *or* 2 *referenced*

Following the IEEE definition above, requirements only describe the conditions or capabilities of the solution (the what), but not the approach how they will be provided (the how). Thus, the requirement and the form of the solution should be considered separated although obviously interlinked. Furthermore, there is a differentiation between requirement and documented requirement. In addition to the capturing, the documentation of requirements is one of the main challenges of RE, giving this differentiation a high importance [23].

5.3.2 Types and Structure

As indicated by the definition above, requirements can be related to capabilities *needed* by certain stakeholders, or capabilities that have to be *provided* by a system. Therefore, requirements in the system development process can be assigned to two distinguished areas: the problem domain and the solution domain, as shown in Fig. 5.4. The problem domain includes the needs and business goals for system development and their formulation into stakeholder requirements, without preselecting any specific solution characteristics. The solution domain contains the system requirements describing the targeted functionalities of the solution and subsequently the architectural design, which specifies how the solution will meet the system requirements [11].

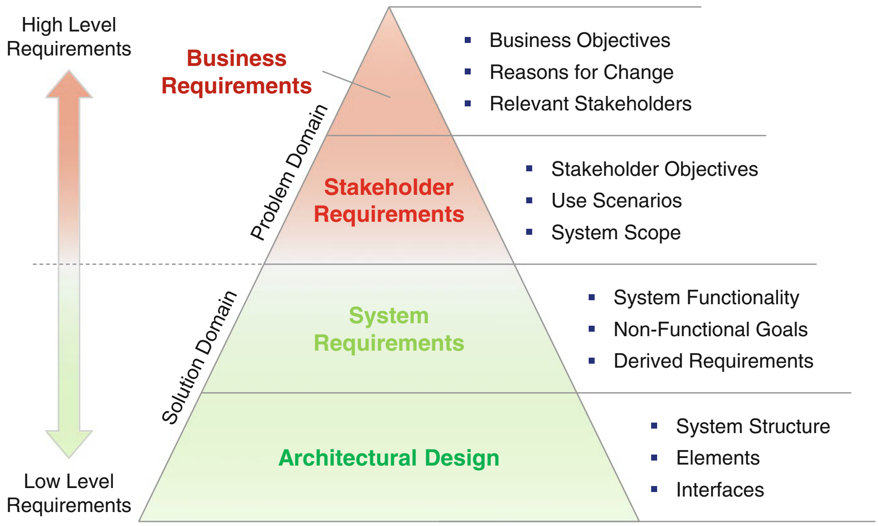


Fig. 5.4 Types of requirements

Thus, requirements are transformed into different types and structured into different groups along the development process. Depending on the specific viewpoint, different layers of requirements can be distinguished. A widely accepted classification is given in Fig. 5.4. Prominent distinctions, which can be found both in literature and practical usage, are business requirements and stakeholder requirements for the problem domain and high-level/low-level system, constraint and architectural requirements for the solution domain [24, 25].

5.3.2.1 Business Requirements

Every development project has to be aligned with the business context in which the resulting system will be implemented. This context and the constraints imposed by it are described by business requirements. They are derived from the business goals or objectives of the organization, which interpret the underlying business vision. Methods for documentation of business requirements can be business process model and notation (BPMN) or data flow diagrams, showing the difference between "as-is" and "to-be" business processes [26]. The extent to which a system supports the business requirements and facilitates an organization in achieving its goals is a key success factor. Therefore it is important to also identify all stakeholders connected to the system and document their requirements [27].

5.3.2.2 Stakeholder Requirements

After having identified the general business requirements for system development, it is essential to determine the expectations that people or organizations have towards the abilities of the intended system. These expectations are formalized in stakeholder requirements. Stakeholders are all persons and institutions involved in the development, construction and the use of the system. Pohl states [20]:

> A Stakeholder of a system is a person or organization, who has a potential interest on the future system and therefore normally makes demands on the system. In doing so, one person can represent the interest of several persons or organizations, that means one person can assume more than one role.

Stakeholder involvement can be direct, like from the users of the intended system, or indirect, like from the investors financing the development.

An important and difficult step in system development is determining what the stakeholders actually expect from the system. It is a challenge for RE to identify and document previously unknown requirements for system development. This is often because stakeholders cannot communicate the entirety of their needs. In addition, the information they provide may also be incomplete, inaccurate and self-conflicting. The responsibility of completely understanding the stakeholder needs thus falls on system developers. However, considering the complexity of modern systems and the volatile environment they often operate in, some requirements are not only unknown, but possibly unknowable due to rapid changes or limited access to information [28]. Some requirements become only apparent when the system evolves and must be incorporated later. In general, stakeholder requirements are derived from the statements of need, using various methods like use scenarios, and are often stated in non-technical terms normally not adequate for design purposes. However, they do provide the measures of effectiveness by which delivered end products will be judged by that stakeholder. Thus, stakeholders are "*the actual information providers for aims, requirements and boundary conditions, whose management is a factor of success for the satisfaction of the customer needs*" [24]. Stakeholder requirements conclude the description of the problem domain towards what the system is expected to do and need to be translated into technical requirements appropriate for system development [29].

5.3.2.3 System Requirements

Once the problem domain has been described, it is necessary to state how the to-be system will address the stakeholder requirements, without limiting the development to any specific design at the beginning. Therefore, the first step in the solution domain is to describe the targeted system behavior in terms of conditions or capabilities of the envisaged solution. This can be done through developing system models describing functionality and then documenting system requirements that capture the vision of the customer in technical terms, enable the definition of the scope of the system and allow estimating the cost and schedule required to build the system. The typical number of requirements for a large system is in the order of 50–200 system-level requirements [30]. These requirements are used to verify the system after development and can be grouped into functional and non-functional requirements.

Functional Requirements

The functionalities that are expected from a solution, i.e. the useful capabilities provided by the system, are described by functional requirements. They specify: (1) the necessary task, action, or activity that must be accomplished, or (2) what the system or one of its components must do [30]. Functional requirements are sometimes called behavioral or operational requirements, because they describe the inputs (stimuli) to the system, the outputs (responses) from the system, as well as the behavioral relationships between them [31]. The document used to communicate the requirements to customers, system, and software engineers is referred to as a functional specification.

Non-functional Requirements

Non-functional requirements define how well the functional requirements must perform by describing the characteristics of the system independently of its functional goals. Thus, they are also called quality attributes, and comprise aspects like usability, likability, performance, reliability or safety; e.g. the time required to execute a function can be a non-functional requirement [32]. During RE, non-functional requirements will be interactively developed across all identified functions based on system's life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements [30].

Derived Requirements

The nature of a system is the composition of elements on different levels that work together to produce a desired result. The system is the highest level and can be divided into subsystems that provide distinct functionalities. The subsystems are made up of tangible and intangible components. Therefore, the requirements on system level have to be refined to derive lower level requirements describing the functionality for subsystems and components of the system [11]. For subsystems, this includes the necessary interfaces to the system level, while components can be seen as an implementation of subsystem's requirements.

A derived requirement is thus either a requirement that is further refined from a higher-level derived requirement or a requirement that results from choosing a specific implementation for a system element [33]. If the system architecture is roughly known in advance, low-level requirements can sometimes be directly derived from stakeholder requirements.

5.3.2.4 Architectural Design

The second layer in the solution domain is the description of the architectural design of a system. It identifies the different system elements and shows how they work together through their relationships to meet the system requirements [34]. In the case of complex systems, this can be the interaction between product and services, or between software and hardware. Thus, a difficult challenge in system development is finding and defining interface requirements. Interface requirements analysis identifies physical and functional relationships among system elements and between system elements and the system environment.

5.3.2.5 Constraints

Constraints do not directly describe required functionalities of a system, but state conditions for their implementation. They can address the system design, realization and application and have to be analyzed for cost, benefit and impact. Constraints expressed by stakeholders should already be addressed in the system requirements. Design constraints limit the alternative architectures, which can be envisaged to satisfy the system requirements. The system will have to comply with constraints such as the use of a specific platform (hardware/software) or development cost and time [5]. Constraints originating from the setting in which the system will be used includes regulations and standards or labeling requirements and requires proof of compliance—for example, through conformity assessment, including certification. However, if the number of constraints is too high, it can make the development of a system impossible [30].

5.4 Requirements Processing

As explained in Sect. 5.2 and illustrated in the V-Model (see Fig. 5.3), RE cannot be seen as an activity that is conducted prior to system development. Especially when using approaches like CE, incremental development or prototyping, requirements can change during the whole development process and even in later life cycle phases [35]. Thus, in every stage of the development process, the intermediate results have to be tested against the original requirements, finally validating the system itself. The general RE process can be divided into sequential requirements development activities and cross-sectional requirements management tasks, as illustrated in Fig. 5.5.

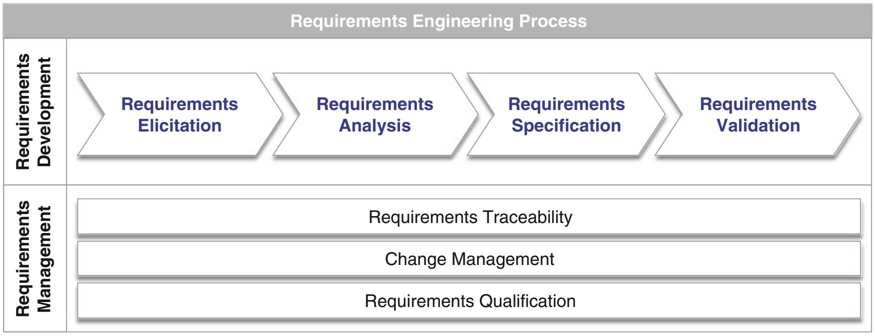


Fig. 5.5 Activities in the requirements engineering process

The following sections will describe the activities involved in discovering, documenting and maintaining a set of requirements for the development of a system in the RE process.

5.4.1 Requirements Development

Requirements development has the purpose to elicit, analyze, and establish stakeholder and system requirements. It includes the identification of requirements for a system, checking the requirements expression and conflicts, documenting the requirements in a specification and validating the documented requirements with the stakeholders [36]. The underlying steps of elicitation, analysis, specification and validation are explained below.

5.4.1.1 Requirements Elicitation

The goal of requirements elicitation is to determine the relevant requirements for a development project. They should enable the creation of a system that brings the highest possible benefit for the goals of the stakeholders. Thus, it has to be ensured that everyone who may be affected by a system is consulted during requirements elicitation. It is not easy to identify all stakeholders of a system and many of the requirements can vary or even be opposite and contradictory. After identification of the stakeholders, various techniques can be used to capture the requirements from the different groups involved, including scenario exploration, interviews, questionnaires and many more [17]. The nature of the requirements to be elicited may vary from small changes to an already existing solution to unclear ideas about solving a totally new problem. Therefore, the focus of requirements elicitation is the improvement of precision, accuracy and variety of requirements.

5.4.1.2 Requirements Analysis

Requirements analysis is used to evaluate and improve the quality of the elicited requirements. A good requirement describes something that is "*necessary, verifiable and reachable*" [37]. Regarding to this definition, a requirement is *necessary*, if the disappearance of this requirements challenges the fulfillment of a contract or the functionality of the solution. A requirement is *verifiable*, if it is clear how to test and approve it by reading the specification. And a requirement is *reachable*, if it is reliable, not in contradiction with another requirement and compatible with all restrictions. Thus, the discovery of hidden/latent interrelations between requirements or missing assumptions during analysis helps to remove obstacles to requirements satisfaction. Furthermore, conformity towards characteristics like ambiguity, inconsistency or incompleteness is assessed. Misunderstandings or questions appearing during the analysis could become the foundation for a second round of elicitation.

5.4.1.3 Requirements Specification

Without an adequate structure in the documentation of requirements, it is not possible to provide the developers with the information they need for creating a complex system. A requirements specification provides a complete description of the functionality of the system to be developed. It also includes non-functional requirements describing constraints. The documentation will typically be structured in a hierarchical way, providing sections and subsections for different levels of requirements [9]. The hierarchy can be based on a modeling approach used for the system development, e.g. usage scenarios or data flow diagrams.

5.4.1.4 Requirements Validation

In systems engineering, validation confirms that the requirements specification meets the needs of the customer and other identified stakeholders. Requirements validation is used to confirm the completeness and correctness of the determined requirements, in order to ensure that the documented requirements accurately express the stakeholder's needs [11]. Therefore, the requirements have to consider all stated business objectives and must be expressed clear and understandable to be able to identify any missing requirements. Requirements validation evaluates the specification against informally described or undocumented requirements. Therefore, the stakeholders have to be involved in the review of the requirements during validation.

5.4.2 Requirements Management

Requirements are used to control the system development and validate the final product, consequently RE has also a strong interdependency with management issues. Requirements management has the purpose to manage requirements of the development project and system components and to ensure alignment between those requirements and the project plan and work results. Typical cross-sectional activities are tracing requirements along the different levels of systems engineering, the planning, monitoring and controlling of changes and qualification of the development results against the requirements input [36].

5.4.2.1 Requirements Traceability

During RE, it is important to understand how the different types of requirements and the system design are connected and transformed into each other. Requirements traceability links lower-level requirements with the higher-level requirements they originate from, so that RE can trace each single requirement to its information source [38]. This enables both assessing the effect of changes to upper layer requirements (e.g. stakeholder requirements) to system development as well as to check if every system element, subsystem or component is linked to a specific stakeholder requirement, or if it can be removed. Finally, the link of all requirements to lower layers and qualification test can be analyzed. Thus, it is possible to assure that all stakeholder requirements are met by the system and will be tested, as well as to examine the progress of system development. Without requirements traceability, neither the progress of RE, the impact of changing requirements or the cost-benefit of certain requirements can be analyzed.

5.4.2.2 Change Management

It is hardly possible to avoid changing requirements when developing complex systems. Due to an alternating environment or varying stakeholders, changes may happen anywhere along the system life cycle and have an impact on systems engineering [35]. Change management ensures that the modified requirements are fed back into the development process, so that the system can further fulfill its intended purpose [39]. The requirements specification for a system contains both stable and volatile requirements. The volatile requirements can further be classified into mutable requirements that will follow changes in the system environment and emergent requirements, which emerge from a system when it is designed and implemented [40]. In general, technological and non-functional constraints are more volatile than high-level and functional requirements [41].

5.4.2.3 Requirements Qualification

Qualification, also termed verification in systems engineering, is the evaluation of whether a system complies with the specified requirements or not [42]. Requirements qualification is conducted to confirm that the design, components and final system fully satisfy the documented requirements. Thus, it is a strategic procedure along the system development process. According to the different objects targeted by qualification, also the activities applied differ from requirement reviews, design inspections, component tests and trials etc., to allow defects in the system to be detected or prevented, where a defect is a departure from requirements. Qualification has to start early, in order to avoid late design changes and rebuilds [11].

5.5 IT Support

As described above, RE aims at developing a precise and effective specification of stakeholder and system requirements. In order to carry out the process as efficient as possible, dedicated software tools have been developed to support requirements development and management. These tools aim at supporting different parts of the RE process, e.g. carried out in spatial distributed environment for complex systems. Many of the tools are designed for a specific purpose, being proprietary and oriented toward a specific environment. In a survey, Carillo et al. [43] assess and compare the usability of various RE tools for different use cases according to the ISO /IEC TR 24766:2009 report.

The analysis accordingly divided the tools into the following categories: (1) elicitation, (2) analysis, (3) specification, (4) modeling, (5) verification and validation, (6) management, (7) traceability, (8) other tool capabilities, and (9) price. The main outcome of this survey is that only a few of the tools really support interoperability by allowing sharing, communicating and cross-collaboration across the tools, and also only a few supports data federation, which helps to reduce multiple data storage and the need of transferring data from one system to another. In the following, the concept of a requirements library, the coverage of the whole system life cycle and the usage of a standard format for exchanging requirements are presented as key issues for systems engineering. The functionality of IBM DOORS is described as a current tool in the area of requirements management.

5.5.1 Library Concept

The development of complex systems relies on a large number of requirements, which need considerable effort for elicitation and documentation. Reusing requirements can thus save time and resources, as well as help to increase quality in RE [44]. The associated improvement in efficiency can contribute to reduced time to market and better system quality. In order to reuse requirements between different development projects, a consistent structure is needed to control and manage the requirements. Activities like finding, selecting and maintaining requirements require high-level grouping and classification to organize the documented requirements. A requirements repository implemented in an IT tool should therefore feature a defined structure in the sense of a library. By grouping the requirements in a hierarchical way, a tree structure can help to decompose system requirements to subsystem and component requirements. This also supports to identify missing requirements, conflicts and inconsistencies, as well as to create testable sub-requirements [9].

5.5.2 Life Cycle Requirements

Requirements and constraints can originate from all phases in the life cycle of a system, such as design, production and usage. However, they are often not actively documented and thus reduce the quality of the system along its life cycle. A holistic view is needed in order to collect all relevant requirements from each life cycle phase. While the early phases of system development are supported by dedicated tools for systems and RE, product data management tools support a more general handling of the workflow and processes. CAD systems are used to visualize and exchange geometrical data with manufacturing and assembly. However, these systems are usually not well integrated and not interoperable [7]. Therefore, IT tools should help to address each life cycle phase individually and make requirements from different sources available to the developers to improve system quality and reduce cost. For gathering and documenting life cycle requirements, the relevant stakeholders in each life cycle phase have to be involved here.

5.5.3 ReqIF—A Standard for Requirements Exchange

In order to exchange requirements between the different stakeholders and tools in systems engineering, a standardized format is needed. The requirements interchange format (ReqIF) provides an XML schema for the exchange of requirements, along with their associated metadata and status. The ReqIF XML file has the root element "REQ-IF", containing information about the file itself and the enclosed data types and requirements, e.g. in relation to a specific development project. The requirements are stored in containers with user-defined attributes, called specification objects (SpecObject). The data type of an attribute can e.g. be an integer number (as used to specify the requirement ID) or a text string (as used to name the author of the requirement), while formatted text or embedded objects like images needed to define the requirement are included in XHMTL. The relationships between these objects are described as "SpecRelations" with additional attributes. Using the defined relationships, a structured representation of SpecObjects is created in hierarchical "Specification" trees, which allow multiple references on the same SpecObject.

ReqIF has its roots in the generic requirements interchange format RIF, defined in 2004 by HIS, a consortium of German automotive manufacturers. In 2008, this format was transferred to ProSTEP iViP e.V. and further developed by a project group responsible for international standardization. The revised version was handed over to the Object Management Group (OMG) in 2010 as Request for Comment, where the acronym was changed to ReqIF. The format was recognized as a standard by OMG in 2011 and revised in 2013 to the current version 1.1 [45].

5.5.4 IBM DOORS—A Requirements Tool

A common tool for requirements management is IBM Rational dynamic object oriented requirements system (DOORS). It is an object oriented requirement system developed by Telelogic, but currently provided by IBM. DOORS supports optimizing requirements communication, collaboration and verification [46]. It is designed as a requirements management application applicable both within a company as well as within the supply chain. The main industrial areas of application are in automotive and space and aviation.

DOORS is based on a proprietary database, is scalable and allows generation of UML models. It supports team collaboration by providing a requirements management system with a centralized location. Requirements related to design items, test plans, test cases and other requirements are linked to each other, and thus the tool offers a high degree of traceability. It also offers a test environment in which the user can manually test the requirements to different cases and finally, it also helps to manage changes to requirements with either a simple pre-defined change proposal system or a more thorough, customizable change control workflow with Rational change management solutions [47].

The DOORS Next Generation also supports better collaboration and team work in spatial distributed environments [46]. According to the assessment carried out in the survey of Carillo de Gea et al., DOORS is also supporting the elicitation, analysis as well as modeling sub-process of the RE, but is mainly designed for the specification [40]. The use of a proprietary DB reduces the possibility to combine it with the use of other tools, but on the other hand side the support of deriving UML models supports an efficient development process.

5.6 Case Studies

This section presents two case studies demonstrating typical usage of the described RE methodologies and tools in different domains and applications. Based upon the cases from Indesit Company and Volkswagen [48], the requirement life cycle is explained and it is shown how RE tools are integrated into the systems engineering concept.

5.6.1 Indesit Company Home Automation Case

Indesit Company is a world leader in design and production of household appliances and homecare devices and it represents a traditional manufacturing firm. Indeed, it founded its business on products so far, addressing traditional home areas, from cooking, to washing and drying, dish care, and cooling. Indesit Company is currently organized in a traditional hierarchical structure and adopts a product-oriented development process. Collaboration with other organizations is focused mainly on the design and development of innovative components for new products, such as co-design with the design and supply-chain and collaboration with supplier to reduce production time and cost [49].

Recently Indesit Company has started looking at Product-Service concepts to innovate its solution portfolio, create new business opportunities and enhance sustainability [50]. It means introducing a new service-oriented approach and the shift from product lifecycle management (PLM) to service lifecycle management (SLM) [51]. It means to manage not simply the product but also services and the necessary infrastructure along the value chain.

The case study focuses on the application of RE techniques to support such innovation process within the company. Indeed, at the beginning Indesit Company didn't know how to move from a product-centred to a service-centred view implied in shifting from production, management and sales of washing machines (WM) or fridges towards a set of services connected with product use. In order to face such a challenge, Indesit Company applied a structured methodology combining some RE techniques with functional analysis to explicate the requirements of creating new service-product solutions. In particular, the business use case (BUC) analysis was adopted to provide clear and easy-to-read models to conceive the new service offer, formalize the TO-BE scenario and elicit the requirements to shift from PLM to SLM. In particular, the case study focused on a flagship product for Indesit Company: WM. This product has been analysed in the current use and its process was projected into the new service-oriented perspective. Table 5.1 describes the Indesit Company case comparing the current and desired situations; the last row contains the actors involved and future actors are indicated in parenthesis.

5.6.1.1 The Adopted Methodology

Indesit Company was supported in moving towards product-service management by a structured methodology consisting of 5 main steps:

Step 1. Investigation of the AS-IS business scenario by adopting Participatory Design techniques and considering both the design process and the service use. Participatory Design directly involved the business end-users as well as the customers to depict the actual situation that is strongly product-oriented. It aimed to understand how the actual product is designed, how the product is considered by the market and valued by the customers, and

Table 5.1 Indesit company case description

Business use case	"Washing machine use at home"
Current functionality	1. Washing clothes: inserting soap, inserting clothes, selecting washing program, washing clothes, extracting clothes, drying clothes
Desired functionality	1. Support to current functionalities: automatic soap loading, automatic selection of the best washing cycle, etc.
	2. Improvement of the washing safety and quality: control of washing results, damage and risks, noise reduction, etc.
	3. Provision of new controls of the WM functioning:
	(a) Analysis, provisioning and optimization of energy consumption, water consumption, soap consumption, etc.
	(b) Monitoring the users' habits
	(c) Provision of new services oriented to the home ecosystem
Stakeholders/ systems	Consumer, home (Indesit Marketing Staff, Indesit Service Centre, Emergency Centre)

how the design and supply-chain intervene. The dual viewpoint (business-driven and market-driven) assured a wider investigation with regard to the design process and the service level of use;

Step 2. Definition of the AS-IS use cases and formalization by combining functional analysis and BUC analysis. Functional analysis allowed obtaining a schematic representation of the product-service architecture and a simplification of the real world, which fully represent both product and service modules and made the analysts choose the desired level of detail from time to time. Subsequently, BUC analysis allowed identifying the involved actors, their roles and their main activities;

Step 3. Definition of the TO-BE process and formalization of the TO-BE business scenario. Functional analysis supported also the definition of the new service-based module in a graphical way and the connections between new and old modules. After that, functional diagrams were shifted into BUC diagrams and new tools and systems were identified;

Step 4. Elicitation of the business requirements to shift from PLM to SLM. It was done on the basis of the TO-BE BUC diagrams that allowed the main differences between AS-IS and TO-BE processes to be easily identified and the new technological tools and system infrastructures required by the new service-oriented scenario quickly defined;

Step 5. Requirements weighting according to the specific context of use. It adopted Participatory Design to involve the ecosystem actors (within and outside the company) to express a weight for each identified requirement according to the relative importance according to their specific role in the ecosystem. Indeed, the product manufacturer like Indesit Company is usually neither a service provider nor a service market operator, so it needs several partners to carry out some specific activities. In this way, requirements were weighted according to the needs of each network partner; in the case study this approach was extended to the complete industrial chain.

5.6.1.2 Implementation

The AS-IS scenario was analysed by focus groups and interviews involving directly the Indesit Company staff. According to step 1 of the adopted methodology, information was formalized by functional analysis and BUC analysis (black-boxes, functional diagrams, BUC diagrams) to clearly and simply define the main actors involved, the basic flow of actions and the system or tools adopted to do that. Trigger events, assumptions and business rules were also defined according to a structure template. The analysis of the "WM use at home" case considered both the customer's actions and the home actions, which are automatically carried out by the customer's home and/or the WM. The main goal of the case described is obviously washing dirty clothes in a correct, efficient and proper way. The goal can be considered the same for both customer and company viewpoints. Figure 5.6 shows the functional analysis (A) and the BUC diagram at the higher level of detail (B). Each action has been further investigated.

The analysis revealed that the product is conceived and designed always before the related service and almost independently: it means that services are added to an existing product only later on, usually by minor changes, and they are managed as product add-ons by adding new HW-SW elements to the initial product.

The TO-BE scenario focused on the idea of selling services instead of products: Indesit continues to sell products but in a more complex Product + Service perspective. As a consequence, the TO-BE scenario no longer includes a specific home function (washing clothes, washing dishes, cooking, cooling, etc.) but a set of functions creating a home-based system, which is characterized by two emerging trends: energy efficiency and Ambient Assisted Living. These trends are external and completely out of the Indesit control. This means adding to the traditional washing action a list of new functions involving the entire home items. They will be related to efficient energy management, smart and remote maintenance, control of home safety and users' safety at home, mobile information services, household functioning planning and remote scheduling, and new solutions to improve current performances (fast drying, assistive load-unload tools, water reuse system, etc.). In the TO-BE BUC diagrams functions in round boxes represent a different service offered. Each function was than detailed (level 2, level 3 and more). The final result is the design of a home-based system, controlled, monitored and connected to the Internet to be accessible from the web by the customers as well as the companies' operators. The TO-BE ecosystem will be constituted by a company ecosystem: Indesit Company as the WM producer and by some business partners providing the additional services or infrastructures. Figure 5.7 shows an example related to the remote maintenance service.

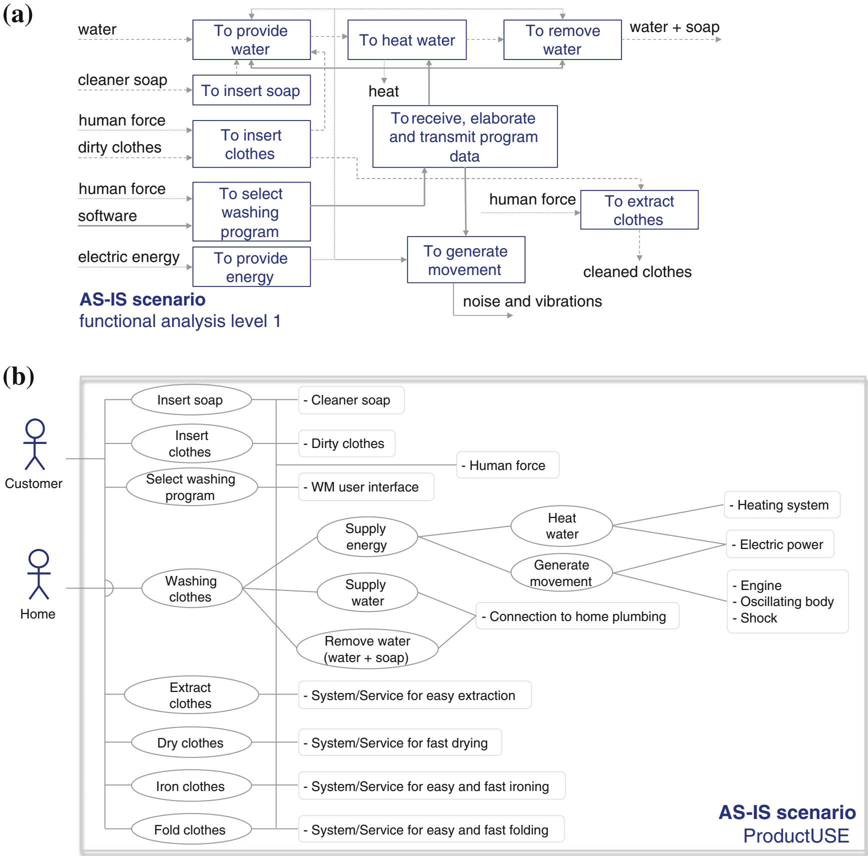


Fig. 5.6 Indesit company AS-IS scenario (examples of diagrams). **a** Indesit company AS-IS functional analysis; **b** Indesit company AS-IS BUC diagram

Requirement elicitation involved directly the key personnel (i.e. managers of the interested areas) from Indesit Company and other three partners, according to their own skill and expertise. In particular, the case study involved the Indesit Innovation Manager, Marketing Manager, Service Manager, IT Manager and the Washing R&D Manager; a set of technical companies providing on-site maintenance, a company providing health and safety services, and a software house developing and delivering mobile applications. Obviously there can be more that one company for addressing large areas (e.g. USA, North Europe, East Europe, etc.). Elicitation was carried out by brainstorming and questionnaires; requirements are collected and then properly weighted according to a 5-point Likert scale. Table 5.2 shows the main results. Three classes of requirements were investigated, related to the product, the software and the SW and the infrastructure. The most relevant requirements were considered as the basis to implement the desired product-service idea (in bold).

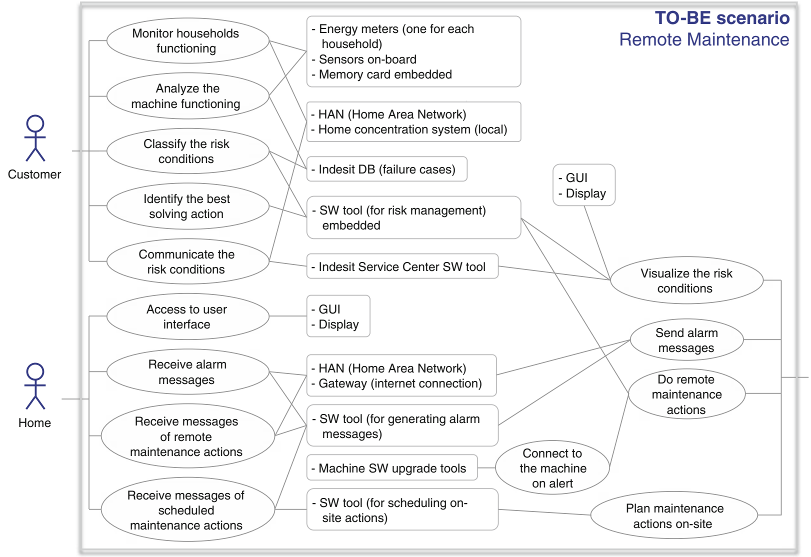


Fig. 5.7 Indesit company TO-BE BUC diagrams for the remote maintenance service (level 1)

The implementation of the proposed methodology for requirement elicitation in Indesit Company and its ecosystem allowed easily defining the requirements for the new Product-Service solution definition and paved the way to the solution design. Indeed, at the beginning the internal business units of Indesit Company as well as the involved suppliers had a lot of difficulties in modelling the final solution, planning the activities to be carried out, identifying clearly the partners' roles and assigning a priority. The above-mentioned case study provides an example of the practical effects of adopting an RE approach in a big manufacturing company (Indesit Company) and in 3 of its suppliers. For Indesit Company it is demonstrated how a big firm, where processes are strongly structured and introducing new processes is extremely difficult, can benefit from RE in supporting new process definition and planning organization also involving external partners. For its suppliers, the example proved how they are supported in process definition and interaction with the leader company is more organized and balanced according to their own skill and expertise.

5.6.2 Automotive Mechanical Case [48]

The trend towards higher system complexity is also influencing the automotive industry. In addition to the rising share of functionality that is realized by electronics and software, the variability of mechanical components necessary to customize cars is drastically increasing. This is a challenge for the OEM's development

Table 5.2 Indesit company case requirement and weights

	Indesit C.	Partner 1	Partner 2	Partner 3	Global weight
Product requirements					
Cleaner soap provisioning	4	1	1	1	1.75
Friendly user interface	5	1	4	5	**3.75**
Water control	4	3	1	1	2.25
Human actions control	5	4	4	3	**4.00**
Electric power control	5	4	3	1	**3.25**
Heating system control	4	2	3	1	2.50
Product movement system control	4	3	1	1	2.25
SW requirements					
Household consumption control	4	5	4	4	**4.25**
Simulating household functioning	3	5	3	4	**3.75**
Simulating energy consumption	3	5	3	4	**3.75**
Risk management	2	5	5	5	**4.25**
Generating alarm message	3	4	5	5	**4.25**
Scheduling on-site actions	2	5	4	4	**3.75**
Detecting user actions	2	4	5	4	**3.75**
Automatic user action classification	2	2	5	2	2.75
Data analysis and elaboration	5	4	5	2	**4.00**
General service management	2	5	5	2	**3.50**
Connection with a home automation system	2	4	5	5	**4.00**
Infrastructure requirements					
GUI (centralized)	5	5	4	1	**3.75**
Local display	5	2	2	1	2.50
Energy meter	2	5	5	2	**3.50**
Home concentration data system (local)	2	5	4	1	3.00
Home area network (HAN)	2	5	5	5	**4.25**
Memory card (embedded)	4	5	5	4	**4.50**
Sensor home network	5	3	5	4	**4.25**
Indesit company DB (centralized)	5	5	3	1	**3.50**
Energy supplier DB (energy consumption)	3	5	5	1	**3.50**
Home risk DB	3	4	5	1	**3.25**
Personal account	1	3	3	5	3.00
Web services	1	5	4	5	**3.75**

departments, which have to coordinate the design and manufacturing of these components on a global scale. A key success factor in the automotive industry is therefore an efficient and effective handling of mechanical component requirements. Volkswagen aims to cope with the underlying complexity by introducing a new product line based requirements management methodology.

Requirements are the basis for the design and manufacturing of every car component. In order to reduce costs and prevent failures, a systematic requirements management approach is necessary. However, many requirements are not component specific, but apply to every assembly. In the case of Volkswagen, these requirements are collected in so called template requirements documents, which are enriched with component specific requirements. In order to improve the reuse of component requirements and to consider lessons learnt, Volkswagen introduced a new requirements management methodology called master requirements document (MRD).

Commonalities and varieties of a component or an assembly within a product line can be specified within a MRD. Requirements documents for every variant can be created using filter mechanisms. Thus, only a single master document has to be maintained, which increases quality and decreases costs. The idea of a central specification document is illustrated in Fig. 5.8, where the MRD represents numerous variants of a component or an assembly. Typically, different file formats are used for the existing requirements documents and different suppliers. Therefore, the MRD methodology for Volkswagen should enable the export of the necessary formats for component requirements.

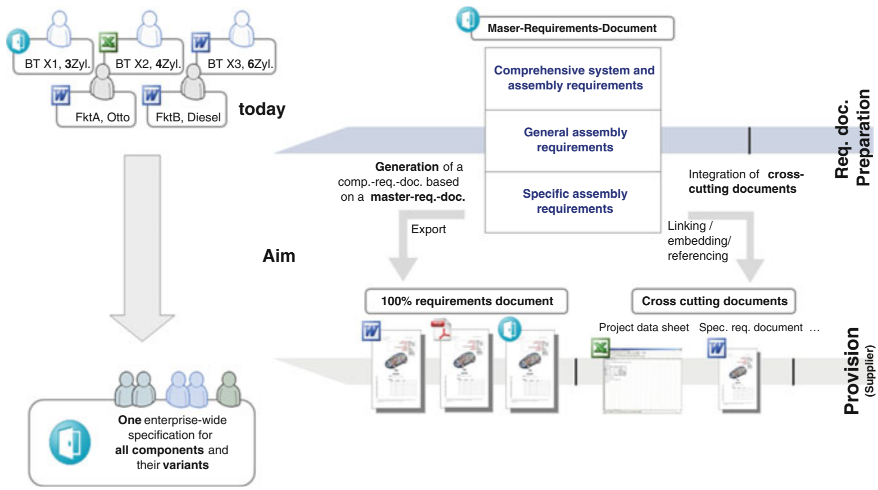


Fig. 5.8 Master-requirements-document methodology [48]

5.6.2.1 Master Requirements Documents (MRD)

Common and variable aspects of a product line can be specified in so called feature models [20]. In this sense, the MRD can be seen as a feature model, integrating component requirements documents (CRD) for all variants of a component or an assembly. Volkswagen distinguishes common and variable parts of a MRD based on three types of requirements, differing by their scope as illustrated in Fig. 5.8.

Comprehensive system and assembly requirements (upper sector in Fig. 5.8) are valid for all components and assemblies of a department, such as the power train development at Volkswagen. As an example, they can provide rules for the verification of every CRD. General assembly requirements (middle sector in Fig. 5.8) specify the corresponding component or assembly. They are valid for all variants of the component and thus included in every requirements document of the assembly as common requirements. Specific assembly requirements (lower sector in Fig. 5.8) specify certain variants of a component or an assembly.

5.6.2.2 The MRD Methodology

In order to manage requirements and create requirements documents, the MRD methodology uses a process with four distinctive steps: (1) generation of the MRD's and their storage in a central database, (2) derivation of CRD's from the existing MRD's, (3) inspection of the derived CRD's and (4) integration of lessons learned and new requirements into the MRD's in a coordinated change process. According to this process, each CRD has to be connected to an associated MRD. Thus, for the generation of MRD's, the existing requirements documents and templates for components and assemblies in a department have to be migrated into the MRD. Similar requirements for the different variants of a component have to be removed and the quality of the requirements has to be checked. The remaining requirements are grouped into the three different types defined in the previous section and complemented with attributes for the different variants, in order to enable the generation of single CRD's. It is possible to access the different groups of requirements through filtered views of the MRD, with the possibility for the designers of creating additional personalized views.

When a designer wants to create a CRD, he can select the MRD for the type of component he would like to specify and generate a personalized view based on attributes for the three requirements groups. Specific assembly requirements can be modified, added and removed to create the final CRD. This CRD can be communicated to the component supplier and is reviewed by editorial staff, which assesses the impact of the changed requirements on the MRD. For instance, changed requirements can be moved to system or assembly requirements, requirements template can be updated or requirements can be added or removed. Changes and adjustments based on CRD reviews are coordinated by the editorial staff. All stakeholders, such as designers and domain experts affected, are instantly informed of the changes.

5.6.2.3 Implementation

Volkswagen documents the MRD methodology in a tool independent handbook. However the actual implementation is based on the IBM Rational DOORS requirements management tool (see Sect. 5.5.4). Access of all requirements documents for the mechanical development departments is enabled by a central data repository, which features simultaneous connections and collaborative work on MRD's in the Volkswagen Group. Role management, filters for different views, versioning and interfaces for other tools are provided by DOORS, supporting the MRD methodology. A user manual helps the end users by explaining the application of the methodology and a support team can be contacted by the designers for the realization of the MRD methodology.

Volkswagen has successfully implemented the MRD methodology to manage requirements and create requirements documents for both simple and complex assemblies. The MRD's help to reuse requirements and consider lessons learned that increases the overall quality of requirements documents. The underlying effort could also be reduced by 33 % according to first rollout experiences. While the effort for identification and usage of known requirement documents could be reduced by 3 %, the effort for adaption and editing is 20 % less. Effort for review and supplier coordination could be shortened by 5 %, respectively 10 %. In contrast, effort for MRD management has only increased by 5 % Furthermore, communication between stakeholders from different domains, such as the designers of Diesel and Otto engines, and from different brands, such as VW and Audi, could be improved.

5.6.3 Results and Discussion

The two case studies show how requirements are developed and managed in industry and what RE concepts and tools are applied. While the Indesit Company case is more focused on the life cycle responsibility and the integration of PLM and SLM, the Volkswagen case concentrates more on managing the rising complexity of systems.

The case of Indesit Company illustrates that the shift from system design and realization only, towards supporting additional life cycle phases (in this case the usage phase) leads to a drastic change in business requirements, which in turn affect stakeholders and system requirements. Furthermore, the system is extended by life cycle services that bring new stakeholders into the RE process. Thus it is necessary to involve more partners deeper for RE than before. Instead of designing the system first and then the services, an integrated development of system and services is necessary. Indesit Company has addressed this challenge by formulating the new business requirements through analysing business scenarios and use cases in collaboration with key partners and the customer. However, the identification of the relevant stakeholders and their involvement in later development phases is not discussed.

In the Volkswagen case, the rising complexity of systems is exemplified by describing the growing number of component and assembly variants caused by the customization of cars. These variants lead on the one hand to a drastically increasing overall number of requirements, of which on the other hand many are redundant or maybe contradictory. This causes a challenge for requirements management to support the alignment and reuse of requirements on different levels and between stakeholders. The MRD methodology addresses the standardization and integration of requirements documents between the component and assembly variants by providing a central repository and filtering mechanisms based on IBM Rational DOORS. However, the approach is momentarily limited to mechanical components only.

5.7 Conclusion and Future Perspectives

The theoretical discussion and the case studies show that the scale and complexity of the objects targeted by systems engineering is constantly growing. Fast technological changes and reduced life cycles demand for an even faster development of systems. Current methods and tools for system development, in particular for RE, do not provide full support for these new challenges. In traditional RE scenarios for conventional systems, the stakeholders are generally aware of their needs. Often, a specific functionality is requested and the system is developed on the basis of formalized requirements by a single enterprise. However, the identification of requirements for the specification of innovative and complex systems poses additional challenges. Complex systems are more innovative and are individually configured for a customer's problem as a one of a kind solution. This makes it harder for the stakeholders to generate the necessary creativity to define their needs towards the solution. Sometimes the problem itself cannot even be described in detail and the requirements are not stable.

Additionally, it has been shown that the realization of complex systems usually requires the temporary collaboration of a multitude of stakeholders from different domains, such as hardware, software and services. Besides the customer/user and the system integrator, there are stakeholder groups for the system components, life cycle services and system environment, each with their own objectives and context. The stakeholder environment grows in size as well as in complexity, leading to various factors to be considered. Stakeholders for the system as a whole may have limited knowledge of the needs and constraints for the individual components, and vice versa. This induces a change in the stakeholder environment from quasi stable and simple socio-technical systems to a more complex and instable dynamic variation. Current RE approaches are not able to handle the large number of different and conflicting requirements without exponentially increasing time and cost, as contradictions and interdependencies have to be assessed for a large number of requirements in various domains [52].

Systems engineering is evolving from a centralized development process for individual systems and components towards the orchestration of distributed software, hardware and business processes for a common purpose. This requires the identification of new interoperability requirements, which describe organizational, technical and management prerequisites for the system realization. New RE concepts and techniques will have to support two main aspects:

1. Collaboration and interoperability between stakeholders and system components from different domains, especially hardware, software and services.
2. Management of unstable and unknowable requirements, taking into account information from all system life cycle phases.

The integration of system components from different domains leads to collaboration and competition between previously separated branches like automotive and IT. The stakeholders from different domains needed for the realization of complex systems typically have their own specific development methodology, standards and even "language". RE needs to support the "translation" of requirements between the domains to enable a common understanding of the desired system. Furthermore, the value chains have to be configured so that the system can be adapted to changing requirements through life time services, even in the usage phase.

New methodologies need to be developed to support interoperability between system components from different domains and describe the emergent system behavior. They need to be able to identify conflicting, unstable and unknowable requirements fragmented across the different domains for complex systems. Instead of static and approved specifications, methods and tools are needed that can anticipate and represent requirements, which are changing dynamically over the system life cycle and its environment. Within the methodology, a model to describe interdependencies between the tangible and intangible components as well as between the stakeholders should be implemented. This would help to comprehensively identify the emergent system behaviour and provide information about which stakeholder needs what information during system development.

References

1. Corsetti A, Ribeiro EA, Garbi GP, Zanta K, Medeiros M, Loureiro G (2013) Complex systems developed with system concurrent engineering. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 1057–1068
2. Chang W, Yan W, Chen CH (2013) Customer requirements elicitation and management for product conceptualization. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 957–968
3. Kossiakoff A, Sweet WN, Seymour S, Biemer SM (2011) Systems engineering principles and practice, 2nd edn. Wiley, Hoboken
4. Blanchard BS (2012) System engineering management, 4th edn. Wiley, Hoboken
5. Sage AP, Rouse WB (2009) Handbook of systems engineering and management, 2nd edn. Wiley, Hoboken

6. Elgh F (2007) Modelling and management of manufacturing requirements in design automation systems. In: Loureiro G et al (eds) Complex systems concurrent engineering. Springer, London, pp 321–328
7. Nilsson P, Fagerström B (2006) Managing stakeholder requirements in a product modelling system. Comput Ind 57(2):167–177
8. Nuseibeh B, Easterbrook S (2000) Requirements engineering: a roadmap. In: Proceedings of the conference on the future of software engineering, Limerick
9. Hauksdóttir D, Mortensen NH, Nielsen PE (2013) Identification of a reusable requirements structure for embedded products in a dynamic market environment. Comput Ind 64(4):351–362
10. Boehm B, Basili B (2001) Software defect reduction top 10 list. In: IEEE Computer, vol 34(1). IEEE Computer Society, Los Alamitos, pp 135–137
11. Hull E, Jackson K, Dick J (2011) Requirements engineering, 3rd edn. Springer, London
12. Royce WW (1970) Managing the development of large software systems. In: Proceedings of IEEE WESCON, vol 26(8)
13. Baxter D, Gao J, Case K, Harding J, Young B, Cochrane S, Dani S (2008) A framework to integrate design knowledge reuse and requirements management in engineering design. Robot Comput-Integr Manuf 24(4):585–593
14. Christopher DFX, Chnadra E (2012) Analyzing the efficacy of requirements stability based on function point modeling. Int J Eng 1(9)
15. Taheri F, An Duong N (2010) Introducing requirement stability metrics for test case success prediction in RUAG space AB. Rep/Dept Appl Inf Technol 2010:63
16. Fuxin F (2005) Configurable product views based on geometry user requirements. Comput Aided Des 37(9):957–966
17. Laporti V, Borges MR, Braganholo V (2009) Athena: a collaborative approach to requirements elicitation. Comput Ind 60(6):367–380
18. Azadegan A, Papamichail KN, Sampaio P (2013) Applying collaborative process design to user requirements elicitation: a case study. Comput Ind 64(7):798–812
19. Mallek S, Daclin N, Chapurlat V (2012) The application of interoperability requirement specification and verification to collaborative processes in industry. Comput Ind 63(7):643–658
20. Pohl K (2008) Requirements engineering; Grundlagen, Prinzipien, Techniken. 2. Auflage. dpunkt Verlag, Heidelberg
21. Rausch A; Broy M (2007) Das V-Modell XT—Grundlagen, Erfahrungen, Werkzeuge. dpunkt. verlag, Heidelberg
22. IEEE Standard 830 (1998) Recommended practice for software requirements specifications. IEEE Press, New York
23. Versteegen G (2004) Einführung in Anforderungsmanagement. In: Anforderungsmanagement. Springer, Heidelberg, pp 1–37
24. Rupp C (2009) Requirementsengineering und –management; Professionelle, Iterative Anforderungsanalyse für die Praxis, 5th edn. Carl Hanser, München, Wien
25. Wallmüller E (2001) Software-Qualitätsmanagement in der Praxis – Software-Qualität durch Führung und Verbesserung von Software-Prozessen. Carl Hanser, München, Wien
26. Liu X, Akinci B, Bergés M, Garrett JH Jr (2013) Extending the information delivery manual approach to identify information requirements for performance analysis of HVAC systems. Adv Eng Inform 27(4):496–505
27. Hass KB, Wessels DJ, Brennan K (2007) Getting it right: business requirement analysis tools and techniques. Management Concepts Press, Vienna
28. Barnes RJ, Gause DC, Way EC (2008) Teaching the unknown and the unknowable in requirements engineering education. In: IEEE Proceedings of the requirements engineering education and training (REET'08). pp 30–37
29. Martin JN (1996) Systems engineering guidebook: a process for developing systems and products, vol 10. CRC Press, Boca Raton
30. Young RR (2004) The requirements engineering handbook. Artech House, Norwood

31. Ambrósio AM, Guimarães DC, Barreto JP (2007) Satellite simulator requirements specification based on standardized space services. In: Loureiro G et al (eds) Complex systems concurrent engineering. Springer, London, pp 175–183
32. Glinz M (2007) On non-functional requirements. In: 15th IEEE international conference on the requirements engineering conference (RE'07), pp 21–26
33. Hause M, Thom F, Moore A (2005) Inside SysML. Comput Control Eng J 16(4):10–15
34. Haneyah SWA, Schutten JMJ, Schuur PC, Zijm WHM (2013) Generic planning and control of automated material handling systems: practical requirements versus existing theory. Comput Ind 64(3):177–190
35. Lim SL, Finkelstein A (2011) Anticipating change in requirements engineering. In: Relating software requirements and architectures. Springer, Heidelberg, pp 17–34
36. Team CP (2010) CMMI for Development (CMMI-DEV) vol 1. Technical report CMU/SEI-2006-TR-008
37. Ebert C (2008) Systematisches requirements engineering und management; Anforderungen ermitteln, spezifizieren, analysieren und verwalten. 2. Auflage. dpunkt Verlag, Heidelberg
38. Wynn MT, Ouyang C, ter Hofstede AH, Fidge CJ (2011) Data and process requirements for product recall coordination. Comput Ind 62(7):776–786
39. Huang HZ, Li Y, Liu W, Liu Y, Wang Z (2011) Evaluation and decision of products conceptual design schemes based on customer requirements. J Mech Sci Technol 25(9):2413–2425
40. Sommerville I (2011) Software engineering, 9th edn. Addison-Wesely, Boston
41. Van Lamsweerde A (2009) Requirements engineering: from system goals to UML models to software specifications
42. Project Management Institute, Inc (2013) A guide to the project management body of knowledge (PMBOKSM Guide), 5th edn. PMI, Pennsylvania. ISBN 978-1935589-67-9
43. Carillo de Gea J, Nicolas J, Fernandez Aleman JL, Toval A, Ebert C, Vizcaino A (2011) Requirements engineering tools. IEEE Soft 28:86–91
44. Massonet P, Van Lamsweerde A (1997) Analogical reuse of requirements frameworks. In: Proceedings of the 3rd IEEE International symposium on requirements engineering, pp 26–37
45. Ebert C, Jastram M (2012) ReqIF: aeamless requirements interchange format between business partners. IEEE Soft 29(5):82–87
46. IBM (2014) Rational DOORS. http://www-03.ibm.com/software/products/en/ratidoor. Accessed 31 Mar 2014
47. IBM (2012) DXL reference manual
48. Gümmer R, Junk C, Rock G (2013) A variant management based methodology for the requirements-engineering process of mechanical parts. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 109–120
49. Peruzzini M, Germani M (2013) Investigating the sustainability of product and product-service systems in the B2C industry. In: Meier H (ed) Product-service integration for sustainable solutions. Lecture notes in production engineering LNPE 6, Springer, Heidelberg, pp 421–434
50. Favi C, Peruzzini M, Germani M (2012) A lifecycle design approach to analyze the eco-sustainability of industrial products and product-service systems. In: DS70 Proceedings of the 12th international design conference DESIGN 2012, Cavtat, 21–24 May, pp 879–888
51. Peruzzini M, Germani M, Favi C (2012) Shift from PLM to SLM: a method to support business requirements elicitation for service innovation. In: Rivest L, Bouras A, Louhichi B (eds) Product lifecycle management. Towards knowledge-rich enterprises, IFIP Advances in information and communication technology 388 (AICT). Springer, New York, pp 111–123
52. Jarke M, Loucopoulos P, Lyytinen K, Mylopoulos J, Robinson W (2011) The brave new world of design requirements. Inf Syst 36(7):992–1008

Chapter 6
Resolving Interoperability in Concurrent Engineering

Nicolas Figay, Catarina Ferreira da Silva, Parisa Ghodous and Ricardo Jardim-Goncalves

Abstract To face an increasingly competitive environment within a globalization context, and to focus on core high-added value business activities, enterprises have to establish partnerships with other companies specialized in complementary domains. Such an approach, primarily based on optimization of the value chain, is called virtualization of the Enterprise. Enterprises relying on virtualization, subcontracting and outsourcing have to coordinate activities of all the partners, to integrate the results of their activities, to manage federated information coming from the different implied information systems and to re-package them as a product for the clients. The adopted organization, which is considering as well as the internal and external resources, is called "Extended Enterprise". Nevertheless, in such complex emerging networked organizations, it is more and more challenging to be able to interchange, to share and to manage internal and external resources such as digital information, digital services and computer-enacted processes. In addition, digital artifacts produced by enterprise activities are more and more heterogeneous and complex. After characterizing expected interoperability for collaborative platform systems and highlighting interoperability issues and brakes not yet addressed, this chapter describes an innovative approach to build interoperability based on a Federated Framework of legacy eBusiness standards of a given ecosystem. It implies facing important issues related to semantic preservation along

N. Figay
EADS, Suresnes, France
e-mail: nicolas.figay@eads.net

C.F. da Silva · P. Ghodous (✉)
LIRIS, UMR5205, University of Lyon 1, CNRS, Villeurbanne F-69622, France
e-mail: parisa.ghodous@liris.cnrs.fr

C.F. da Silva
e-mail: catarina.ferreira-da-silva@liris.cnrs.fr

R. Jardim-Goncalves
New University of Lisbon, Uninova, Lisbon, Portugal
e-mail: rg@uninova.pt

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_6

the lifecycle of the artifacts and infrastructures required to define and exploit an application. We present two use case studies that apply interoperability strategies.

Keywords Interoperability · Enterprise standard · Organization network · Extended enterprise · Enterprise virtualization · Federation · Servitization

6.1 Introduction

In the advent of a global networked economy, organizations are being confronted with new opportunities and challenges in the workspace and marketplace, trying constantly to shift the boundaries of their operations and collaborations to competitive exploitation of new business models and markets.

To achieve that, one of the difficulties enterprises have to address is the lack of interoperability of software applications to manage and progress in their business. Interoperability is the ability of systems to provide services to and accept services from other systems and to use the services exchanged in this way to enable them to operate effectively together. Although several definitions exist, in short, a service is an information technology (IT) representation of self-contained business functionality.

According to Ford et al. [1], 64 types of interoperability are mentioned in research papers, demonstrating the richness of the interoperability field. Is it needed to agree on a single and precise definition? As stated by Morris et al. [2], "We may never have any agreement on a precise definition due to differing expectations that are constantly changing. New capabilities and functions … continue to offer new opportunities for interactions between systems."

Organizations are looking for new business relationships, and the exchange of information and documents with new partners is often incapable of being executed automatically and in electronic format. This is principally due to problems of incompatibility with the information representation adopted by the software applications they are working with.

A typical situation found in companies interested in joining virtualized environments was identified to be directly related with the companies' previous investments in equipment and software. Usually focused to solve local and particular problems, such acquisitions cause information segmentation and make impracticable the functional integration with third parties, due to the incompatibility in data access and data format representation.

Even inside the same company, very often when it gets a new software application, this cannot be integrated with the other applications already running. This means that although the company achieves new informatics capabilities, the data flow is not automatic and paperless, maintaining a high rate of errors in data exchange due to human intervention. Moreover, while most of the legacy information resources are simple replacements of paper-based documents and databases, new artifacts are no longer simple substitutes of paper-based information by electronic media. They are formalized within a business context, according to complex business rules and holistic system engineering processes in order to support automated operations and exploitation, such as querying, validation, reasoning, simulation, knowledge management, etc.

Therefore, the interoperability problem makes major decisions of IT managers difficult when looking for a new application, where the criteria for choice must be balanced between (1) an application that completely fulfills their needs; and (2) an application for electronic data exchange already compatible and ready to be integrated with the existent computational environment. Yet, even when conformance in data format and access is achieved and verified reliable interoperability of information semantics generally is not.

This chapter is structured as follows: Sect. 6.2 addresses data and semantic models as well as existing standards. Sections 6.3 and 6.4 cover methods to support interoperability and frameworks for resolving interoperability, respectively. In Sect. 6.5 two major case studies are presented, one in the aerospace industry and the other in the building and construction (B&C) domain. Finally, conclusions and outlook close this chapter.

6.2 Data, Standards and Semantics

Today, many proposals are available to represent data models and services for the main business and manufacturing activities. Some are released with International Standards (e.g., International Standardization Organisation—ISO), others are developed at regional or national level (e.g., Association Française pour la NORmalisation—AFNOR), or by independent project teams and groups (e.g., Object Management Group—OMG). Most of the available standard-based models have been developed in close contact with industry, following an established methodology. They use optimized software architectures, conferring configurable mechanisms focused on the concepts of extensibility and easy reuse.

Data modeling, sharing and exchange, reuse of models, automatic code generators, software libraries and conformance testing, together with the possibility to incorporate expertise and knowledge representation, are some of the many challenges to face when working in environments supported by heterogeneous platforms and concepts.

Hence, the use of effective and de facto standards to represent data, knowledge and services has shown to be fundamental in helping interoperability between systems. Some examples of these standards are: (1) the OMG standard, defining interfaces for services of a product data management (PDM) system in a distributed and object-oriented environment; (2) the STEP standard, defining the representation of product data to be managed by the PDM system, or; (3) XML, for structured data exchange using internet.

Thus, the integration and mix of different standards and de facto standards have become the basis to implement a complete and harmonized environment. However, each standard's aims, scope, suitability for the purpose of and the possibilities for integration with others has to be clearly understood by those aiming to adopt them to avoid misuse, e.g., in terms of data representation in UN/EDIFACT, STEP and XML: how they can interact, their benefits, advantage and drawback.

One possibility to solve this problem is to develop and propose a global and unique data model covering all requirements from all users in all sectors and give adequate technical training to industry, but this is not a realistic solution. The huge number of different applications and the divergent interests in the software market will immediately crush this possibility. Even if achieved, all applications would be obliged to adopt it, and none is able to impose that.

For instance, ISO10303—STEP is offering a steady methodology with a set of Application Protocols (APs), which are such unique data models, for product data representation for many of the major industrial sectors [3].

6.2.1 ISO10303 STEP: STandard for the Exchange of Product Model Data

STEP—STandard for the Exchange of Product model data, is an ISO (International Standardization Organisation)/TC 184 (Technical Committee: Industrial automation systems and integration)/SC4 (Subcommittee: Industrial data) International Standard (IS) officially identified as ISO10303, for the computer-interpretable representation of product information and for the exchange of product data. The objective of STEP is to provide a neutral mechanism capable of describing products throughout their life cycle.

STEP publishes a proposal for a methodology for development, implementation and validation of an open architecture for exchange and sharing of product data, together with a set of public data models identified as APs.

During the last years the ISO TC184/SC4 community has been working on several definitions of APs for some of main recognized production system areas, such as the automotive, aircraft, electrical/electronics, shipbuilding, oil and gas, and B&C. Nowadays, there are more than 40 APs registered in ISO.

This standard mainly contributes with worldwide open systems adopting neutral networking communication for product data exchange between heterogeneous systems, both in-house and with third parties. Also, it assists in implementation of system-independent architectures, flexible migration policies, contributing to long-term archiving in paperless and life-cycle maintenance support.

At this moment, the creation of a global model to support large-scale company requirements, i.e. for all the phases of product lifecycle, within all the supply chain and for a whole domain business ecosystem, is one challenge to be addressed by the international scientific community. STEP has been presented as a viable alternative to the current state of multiple, incomplete, overlapped and proprietary data formats, seeking solid and reliable data exchange between partners using heterogeneous systems [4].

STEP mainly contributes to worldwide open systems networking communication of product data for neutral data exchange between heterogeneous systems, both in-house and with third parties, long-term archiving, system-independent architecture, flexible migration policies. It also contributes to paperless and life-cycle maintenance support.

On the electronic business side, there are also valuable proposals to represent documents and business data for most of the areas of electronic activity [5]. Perhaps the most relevant at this moment is the ebXML framework [6]. Also, the Unified Modeling Language (UML) developed by the OMG for specifying, visualizing, constructing, and documenting software system has been proven successful in the modeling of large and complex systems. UML provides users with a modeling language to develop and exchange models with mechanisms to extend the core concepts, independent of particular programming languages and development processes.

However, even if project partners use the same standard language to represent and exchange product data, its semantics, (i.e. its meaning—c.f. ontology definition in the next section) can be misunderstood when not well captured.

In order to collaborate on a distributed project, remote engineers and designers need active help to coordinate their efforts. This coordination involves translation of terminology among disciplines, locating/providing generic analysis services, prototyping services, and product management [7]. With applications in fields such as knowledge management, information retrieval, natural language processing, e-Commerce, information integration or the emerging Semantic Web, ontologies are part of an approach for building intelligent information systems: they are intended to provide knowledge engineers with reusable pieces of declarative knowledge (i.e. providing set of true facts and rules).

The developed technologies for ontologies on the Web, based on RDF-XML and combined with emerging RESTFull technologies, are providing the ground for Linked Data, with ability for application to publish their data in a way they can be linked and queried as distributed semantic graphs on the Web. Such technologies are associated with STEP through OASIS PLCS Data EXchange sets (DEXs) Reference Data Libraries (RDL). It allows giving a controlled access to specific classifications, for a given discipline, a given organization or a given application, to be used for interpretation of STEP data within a specific context, being organizational or discipline.

The next section presents some semantic-oriented languages. These languages focus on services and can also be applied to the product lifecycle, as the latter currently has a higher service component than in previous decades. This trend is known as servitization of products, i.e. focus on services around data more than on data structure. A service is a logical grouping operations, for which are distinguish the service provider and the service consumer. Languages addressed in the following section are summarized in Fig. 6.1.

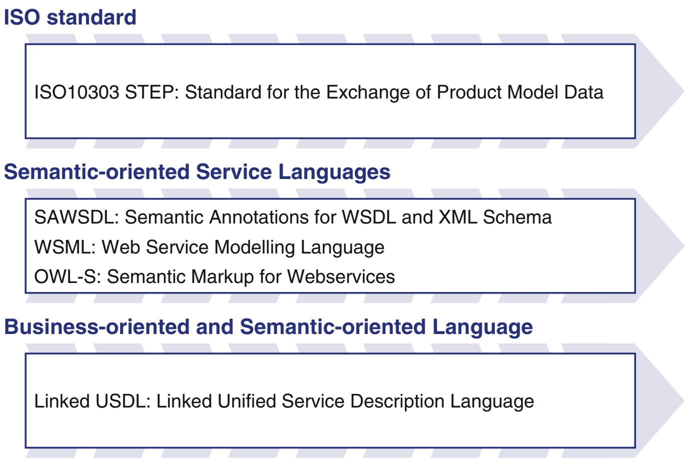


Fig. 6.1 Languages for interoperability

6.2.2 Semantic-Oriented Service Languages

Traditionally, services are described using XML language [8], for instance with the Web Services Description Language, WSDL [9], or its second version, WSDL 2.0 [10]. This language specifies a format to define service interfaces, i.e., the technical aspects of calling web services. It encompasses two different aspects of a service that are its signature, particularly service name and service parameters, and its binding and deployments details, such as protocol and location. Although WSDL 2.0 provides the ability to extend WSDL files, the underlying XML language does not enable to convey precise and unambiguous semantics, i.e. meaning of operations, their typed output or parameters. This means a WSDL file is not enough to manage the whole service contract.

According to Haller et al. [11] determining the semantics for services interfaces means to define the concepts as well as the relationships between them through ontologies. According to Gruber [12] an ontology is a formal explicit specification of a shared conceptualization. Thus, it defines a common agreement on terminology by providing a set of concepts and relationships among those concepts. In order to capture semantic properties of relations and concepts, an ontology generally also provides a set of axioms, which are expressions in a logical language.

Representational techniques being developed for the Semantic Web, an initiative for the web of the future with more intelligence relying on distributed ontologies and intelligent agents, can be used to capture and process semantics. Some of these techniques are grounded on the XML language, bringing complementary language constructs. From the W3C, the Semantic Web Activity group recommends specific languages such as the Resource Description Framework (RDF) [13], RDF Schema [RDF(S)] [14] and Web Ontology Language (OWL) [15, 16]. Particularly, OWL includes three sublanguages: OWL-lite, OWL-DL, and OWL full. The first two, but not the third, correspond to decidable description logics [17]. Decidability implies that fundamental questions about an ontology are guaranteed to be answerable, such as the question of subsumption. A specific class *A* subsumes another class *B* when it is a superclass of this class *B*.

In the domain of Semantic Web Services, the research community has proposed several structured service description languages. Examples of these are Semantic Markup for Web Services, OWL-S [18, 19] and Web Service Modeling Language, WSML [20], which have formal logic semantics groundings. Another outcome in this domain is the Semantic Annotations for WSDL and XML Schema, SAWSDL [21], a W3C 2007s recommendation, which does not have any formal semantics. It is not widely used yet by the PLM community, and remains a challenge for the future.

6.2.2.1 Semantic Annotations for WSDL and XML Schema

The SAWSDL approach [21] proposes a set of extension attributes for the WSDL and XML Schema definition languages that allows description of additional semantics of services described with WSDL language. The SAWSDL specification defines how semantic annotation is accomplished using references to semantic models, such as ontologies. It provides mechanisms by which concepts from these semantic models, typically defined outside the WSDL document, can be referenced from within WSDL and XML Schema components using annotations. For this SAWSDL defines the three extensibility attributes to WSDL 2.0 elements for their semantic annotation, that are a *modelReference* extension attribute (enabling to specify the association between a WSDL or XML Schema component and a concept in some semantic model) and *liftingSchemaMapping* and *loweringSchemaMapping* extension attributes (to transforms XML data into instances of a semantic model and vice versa).

Hereafter, we discuss some limitations and advantages of this approach. Quoting from the example section [22] of the SAWSDL recommendation: "Practice has shown that it is a very hard task to create XSLT or XQuery transformations that take arbitrary RDF/XML as input." As so, to lower schema mappings, they use XML technologies combined with an RDF query language like SPARQL to preprocess the RDF data. Thus, using SAWSDL implies the need to rely on outside software to solve semantic heterogeneities. In real applications, this task is probably assigned to external mediators.

As some OWL sublanguages bring more constraints and expression than RDF, a reference model defined in OWL has to be pre-processed with OWL specific tools as well. Regarding lowering schema mapping, transformations from OWL to XML can cause information loss, since XML is a less expressive language. Nevertheless, SAWSDL is less complex than other service languages such as OWL-S (Semantic Markup for Web Services) or Web Service Modelling Language (WSML) in the sense it only adds three basic constructs to connect XML WSDL representations to outside metadata information. As so, SAWSDL is convenient for applications and domain reference models that do not need the complexity or expressivity of OWL-S or WSML languages. To support SAWSDL, some software was developed, such as Lumina [23] and Radiant [24], both part of the METEOR-S project [25, 26].

6.2.2.2 Web Service Modelling Language

The WSML [27] is a formal language for the semantic markup of web services. It is used to describe a semantic web service in terms of its functionality (service capability), imported ontologies, and interface to enable access. WSML syntax mainly derives from F-logic. It also has a normative human-readable syntax, an XML and RDF syntax. WSML comes in five variants that are WSML-Core, WSML-DL, WSML-Flight, WSML-Rule and WSML-Full.

According to Klusch [28], "A WSML service capability describes the state-based functionality of a service in terms of its precondition (conditions over the information space), postcondition (result of service execution delivered to the user), assumption (conditions over the world state to meet before service execution), and effect (how does the execution change the world state). Roughly speaking, a WSML service capability consists of references to logical expressions in a WSML variant that are named by the scope (precondition, postcondition, assumption, effect, capability) they intend to describe." A specific Service Oriented Architecture (SOA) is proposed in [29] that applies Web Service Modeling Ontology (WSMO) framework and uses a specific execution environment, called Web Service Execution Environment, WSMX [30]. In this environment, they need specific adapters to transform external messages into the WSML compliant format understood by WSMX, and mediators that perform tasks such as translation between ontologies. To support WSML some other software was developed, such as the WSML service editor associated with the WSMO studio [31], WSML-DL and WSML-Rule reasoner and the WSML validator. For instance, the SUPER [32] project uses WSMO as the underlying ontology.

A major criticism of WSML concerns the lack of formal semantics of its service interface and the lack of principled guidelines for developing the proposed types of WSMO mediators for services and goals in concrete terms [33].

6.2.2.3 Semantic Markup for Web Services

Based on OWL, Martin et al. [18, 19] propose OWL-S (Semantic Markup for Web Services) also known as OWL for Services. OWL-S superseded DAML-S [34]. OWL-S intends to add precise semantics to service description. In order to link OWL-S to WSDL some attributes are added to WSDL extensions so that to connect both languages and the generated files. Thus, maps were specified between OWL-S parameters and WSDL message parts.

OWL-S consists in three parts: the service profile, the process model (captured by the *ServiceModel* class) and the grounding (through the property *supports* referring to the *ServiceGrounding* class). The service profile sets out what a service does and is used to advertise the service. The process model aims at describing how the service is used, i.e., gives a detailed description of a service's operation. The grounding provides details on how to interact with a service, via messages. The service profile intends to allow service providers to advertise their service and service requesters, also known as service consumers, to specify what capabilities they expect from the service they need. In OWL-S 1.0, a service profile includes functional parameters that are *hasInput*, *hasOutput*, *Precondition and Effect* (known colloquially as IOPEs), as well as non-functional parameters such as *serviceName*, *serviceCategory*, *qualityRating*, *textDescription*, and meta-data about the service provider. Input and output parameters specify the data transformation produced by processes. Here, a process means a specification of the ways a client may interact with a service. Therefore a process can generate and return new information based on information it is given and the world state. Information production is described by the inputs and outputs of the process. A process can produce a change in the world and the preconditions and effects of the process describe this transition. Preconditions specify facts required prior to the execution of the service. Effects are the expected result from the successful execution of the service.

The semantics of each input and output parameter are defined as an OWL concept formally specified in a given ontology, while preconditions and effects are represented as logical formulas that can be expressed in any appropriate logic (rule) language such as KIF, PDDL, SWRL or SPIN [35].

OWL-S benefits from a large support from the community. Several software and applications were developed and are being developed for this language and ontology of semantic service descriptions, such as the OWL-S editor [36], the OWL-S API [37] and OWL-S service matchmakers, like OWLS-UDDI [38], OWLSM [39] and OWLS-MX [40], to name a few. Moreover, OWL-S grounds its success on existing W3C Web standards such as WSDL and semantic web languages like OWL.

According to Klusch and Fries [41], neither OWL-S nor WSML provide any agreed formal standard workflow-based semantics of the service process model (orchestration or choreography). Alternatively, for abstract service descriptions grounded in WSDL, the process model can be intuitively mapped to BPEL orchestrations with certain formal semantics. In the EU project SUPER, an ontology and extensions for BPEL, named sBPEL, were proposed which allow a process to interact with Semantic Web services [42].

6.2.2.4 Linked Unified Service Description Language

Other service description languages are appearing, such as the Unified Service Description Language (USDL) [43] that also represents the business perspective and its related items (service quality, service level, economic and legal aspects).

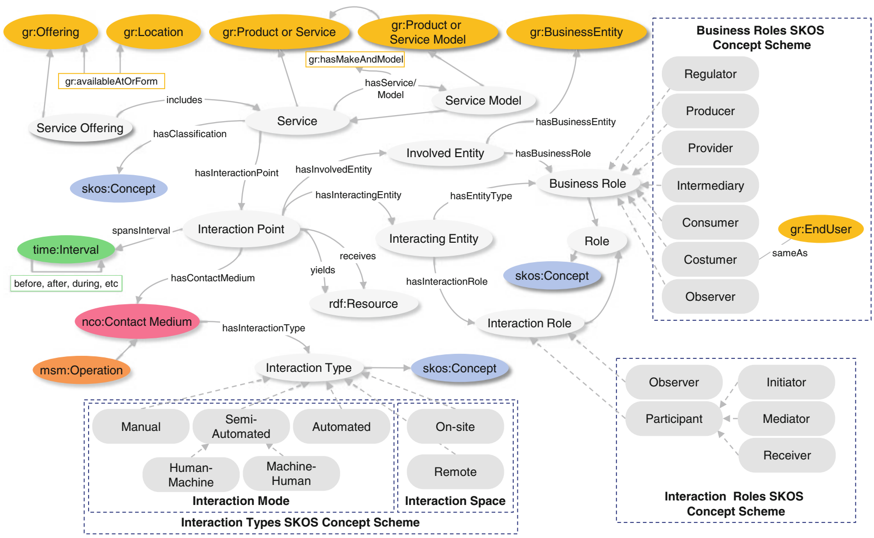


Fig. 6.2 The linked USDL core, from [45]

USDL aims to model service concepts and properties such as service level, pricing, legal aspects, participants, marketing material, distribution channels, bundling, operations, interfaces, resources, etc., with the ultimate purpose of providing a comprehensive view on services. Linked USDL, the USDL semantic version, is a new approach in service descriptions, and is inspired in the Linked-Services concept [44], which are described as Linked Data. These service descriptions whereby their inputs and outputs, their functionality, and their nonfunctional properties are described in terms of (reused) light weight RDFS vocabularies and exposed following Linked Data principles. The term Linked USDL presents the connection it has to linked data and linked services, and is an evolution of the USDL. The use of Linked USDL, and its usage of linked data principles can facilitate service discovery and further improve service composition. The core elements of Linked USDL are represented in Fig. 6.2.

6.3 Methods to Support Interoperability

By nature, all large systems are heterogeneous, i.e., they lack uniformity. Their components were initially developed to address various purposes and evolved towards accretions of different platforms, programming languages, and even middleware. The SOA paradigm enables dealing with such heterogeneous systems in a decentralized way as much as possible. Decentralization helps to obtain loose coupling, one of SOA's key technical concepts along with services and interoperability. We briefly describe these three concepts below.

Loose coupling minimizes dependencies and thus helps scalability, flexibility and fault tolerance. When dependencies are reduced, modifications have minimized effects and the systems still run when some of them are down. When problems occur, it is important to decrease their effects and consequences. Josuttis [46] elaborates on several strategies to apply loose coupling.

The ISO terminology recommendation [47] describes interoperability as the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units. Thus, interoperability enables systems to communicate, understand each other and exchange information. Syntactic and structural interoperability is already set up with transformations, for instance, using standards like XML and XML Schema and associated tools. Syntactic and structural transformations are used to convert schema representations into a target format. However, approaches that target at enhancing interoperability based on structure and on syntax can only produce improvements when a certain conceptual homogeneity exists between graphs to compare. Solving mismatches on the semantic level, i.e. to achieve semantic interoperability, is a complex accomplishment as indicated in the previous section.

Particularly, semantic interoperability is the ability to exchange information and use it, ensuring that the precise meaning of the information is understood by any other application that was not initially developed for this purpose [48]. Semantic interoperability enables systems to process the information produced by other applications, i.e. use it isolated or combined with their own information, in a meaningful way. Therefore, semantic interoperability is an important requirement for improving communication and productivity.

Many SOA definitions mention Web Services, and, although this technology is just one possible implementation strategy to realize the infrastructure [49], it is the de facto standard for current SOA implementations. However, web services related technologies deal with almost exclusively syntactic and structural aspects of information and lack of semantics considerations.

Haller et al. [11] state that the usage of semantic web services and semantic SOA can help overcome the limitations of traditional SOA. This can be done by facilitating the matching of semantically similar operations in different systems, by supporting service mediation through ontology adaptation. This is true for both, process mediation and data mediation, according to the definitions of Fensel and Bussler [49], and by providing the standard Web Services communication mechanisms for system and process-independent communication. To support these tasks and increase the automation in Enterprise Application Integration (EAI), Bouras et al. [50] proposed ENIO, an ontology that permits shared understanding of data, services and processes within B2B integration scenarios, while Izza et al. [51] proposed OSDOI, a framework for EAI evolution using semantic Web Services.

6.3.1 Semantic Mediation

Independently of specific SOA infrastructure or public registries of services, at some moment in the SOA lifecycle it is necessary to match service request descriptions with available service descriptions, in order to verify if the latter corresponds to service consumer needs. This kind of task is common in inter-EAI where it is assumed that a market for services exists and to find the service best suited to the required task is needed, but it can also be present in intra-EAI situations where a company comprises sub-units that evolve individual solutions (even if service-enabled) in partial isolation. To automate this task as much as possible, both consumer and provider service descriptions have to be precisely described, such as within ontologies of services.

Loose coupling usually leads to a situation where only a few fundamental and stable concepts, attributes and data types are defined as a common data model or ontology. However, there will always be ontologies for the same domain created by different communities around the world. Thus, services are described in different ontologies. Therefore, it is necessary to provide the means of finding semantic similarities between them, i.e. by aligning the service ontologies. Mediators can do this task, for instance within an Enterprise Service Bus (ESB), that can help a service call performed by a consumer to find the service provider that can process this request. Josuttis [46] details functionalities of ESB.

Aligning ontologies means discovering a collection of mappings between concepts of these ontologies [52, 53]. Keeping ontology consumer services separated from ontology provider services serves loose coupling.

If we try harmonizing the different ontologies by introducing a common ontology inside the ESB, for instance by merging the input ontologies instead of aligning them, we will easily disable the effect of loose coupling. Moreover, since in dynamic runtime environments the partners, i.e. service consumers and service providers, are not known beforehand, to build a merged ontology during design time does not seem feasible or worthy.

Some approaches try to bring about automation in order to help the complex and tedious mapping task, especially when reference models, such as ontologies, are huge. For instance, CtxMatch-2.1 [54] incorporates a DL-based reasoner to find mappings and to align ontologies. Klusch [28] classifies semantic matchmaking techniques, and their associated tools, as logic-based, non-logic-based and hybrid:

- Non-logic-based matching applies techniques such as graph matching, data mining, linguistics, or content-based information retrieval to exploit semantics that are either commonly shared (in XML namespaces) or implicit in patterns or relative frequencies of terms in service descriptions;
- Logic-based semantic matching of services like those written in the service description languages OWL-S and WSML exploit standard logic inferences;
- Hybrid matching refers to the combined use of both previous types of matching.

Klusch and Fries [41] states hybrid matchmakers, based on syntactic matching techniques, produce better results than only logic-based matchmakers (under non specified conditions), as a result of the first experimental evaluation of the performance of hybrid semantic service matchmakers OWLS-MX [55] and iMatcher2 [56]. From our perspective, however, the choice of the matchmaker depends on the context, particularly on the ontologies and service descriptions at hand. For instance, if only logic-based semantic service descriptions are available, then it seems inappropriate to apply non-logic-based or hybrid matching.

Each of the implemented Semantic Web Service matchmakers supports only one of the many existing Semantic Web Service description formats [28]. Very few matchmakers ignore the structured Semantic Web Service description formats, using monolithic descriptions of services in terms of a single service concept written in a given DL. In such case, semantic matching directly uses DL inference, such as performed by Pellet [57] and Racer [58]. The iSeM matchmaker [59] performs adaptive hybrid semantic IOPE-based selection of OWL-S services and approximated logical reasoning used for evidential coherence-based pruning of multi-dimensional matching feature space.

Currently, Semantic Web service matchmakers (particularly those participating in the S3 competition [60]) perform service profile matching rather than service process model matching. Service profile matching determines the semantic correspondence between services based on the description of their profiles. Semantic matching of service process models, in general, is very uncommon. Naeem et al. [61] propose a kind of partial mapping, based on isomorphism of contracts for workflow and automatic service composition. Unfortunately, the authors do not provide an implementation of their approach.

6.4 Federation of Frameworks for Resolving Interoperability

In this section we describe a federated framework that is composed of the ISO STEP standard, the ATHENA Interoperability Framework and the System of Systems Interoperability. Nowadays, some of the major industrial companies in the world are using STEP to help in the interoperability problem of its manufacturing systems. The ISO STEP standard was described in Sect. 6.2 already. Before presenting the federated framework, the ATHENA Interoperability Framework and the System of Systems Interoperability are described.

6.4.1 ATHENA Interoperability Framework

The operational ability to collaborate is a key success factor for networked enterprises. Interoperability is a necessity for the enterprises involved in collaborations.

The ATHENA Interoperability Framework [62] provides a framework and associated reference architecture for capturing the solutions to interoperability issues that address the problem in a holistic way.

The goal is to reach real, meaningful interoperation between enterprises. ATHENA builds upon the FP5 thematic network IDEAS (Interoperability Development for Enterprise Applications and Software, IST-2001-37368). This network identified the need for a structured approach to collect, identify and represent the current state of the art, vision statements, and research challenges. It defined a framework for capturing and inter-relating this information from many perspectives, including the business, knowledge and ICT layers, plus the cross cutting semantic dimension which captures and represents the actual meaning of concepts and thus promoting understanding. Achieving meaningful interoperability between enterprises implies to deal with all the layers:

- Interoperability at business level is the organizational and operational ability of an enterprise to factually cooperate with other external organizations;
- Interoperability at knowledge level is the compatibility of the skills, competencies, and knowledge assets of an enterprise with those of other, external organizations;
- Interoperability at ICT systems level is the ability of an enterprise's Information System used technologies to be interconnected with those of other external organizations.

The actors involved at each layer, being human or automate, cannot communicate with ensuring that semantics are exchangeable and based on a common understanding to be indeed a means to enhance interoperability. This is reflected by Fig. 6.3.

The ATHENA project adopted an original approach based on a multidisciplinary approach merging several areas supporting the development of interoperability of

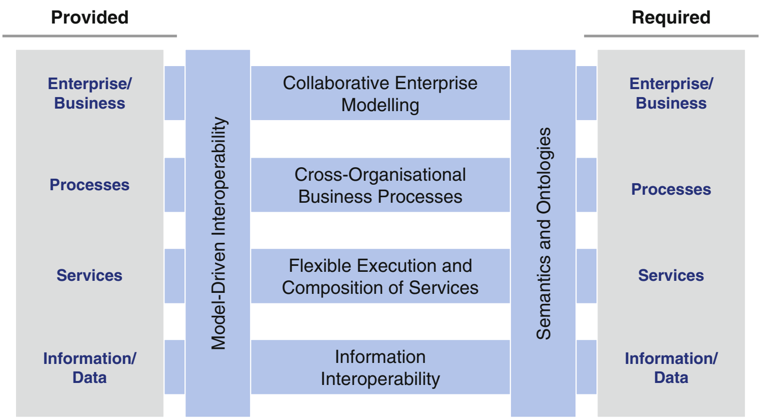


Fig. 6.3 ATHENA interoperability framework

enterprise applications and software: Enterprise Modeling, SOA, Ontology on the Web and Model Driven Architecture.

The aeronautic Industry was involved within ATHENA through business scenarios related to the Networked Collaborative Product development within the supply chain. The objective was to provide experience coming from Product Data Exchange, trying to achieve inclusion of Manufacturing Standards for Product Data Exchange, Sharing and Long Term Archiving—STEP—and associated standardization specifications for PLM services or processes of reference (e.g. VDA Change Management Process).

The integration of the different solutions proposed by ATHENA was difficult and highlighted numerous issues still to be resolved for achieving pragmatic interoperability at an acceptable price to support collaboration in the Manufacturing area [63].

6.4.2 System of Systems Interoperability

The System of Systems Interoperability (SOSI) model, defined in [2], is based on the observation that interoperability in operations cannot be achieved without:

- Activities related to acquisition of a system (program management);
- Activities creating and sustaining interoperability with focus on architecture, standards and Commercial Off-The-Shelves;
- Activities related to operation, based on interactions between systems and with users, including operational support.

Figure 6.4 shows the three sorts of activities that must be further aligned for effective interoperability. The SOSI model defines three type of interoperability: programmatic, constructive, and operational.

Programmatic Interoperability is related to the required cooperation between programs or projects building interoperating systems. Reaching some agreement on

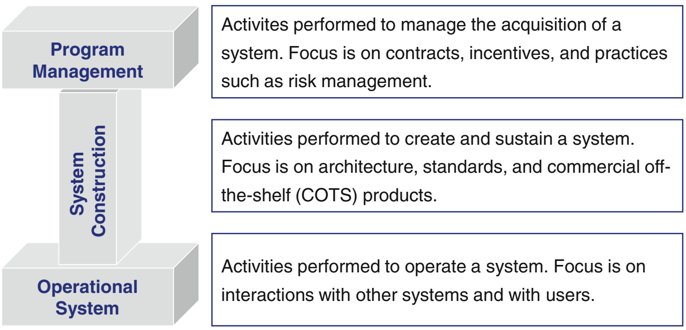


Fig. 6.4 The SOSI model from [2]

requirements regarding the interoperation, the formats of information to be exchanged or shared, and the quality if service expectations and the control of versions of the systems are different ways to achieve it. It is important to share information between people in programs in order to achieve mutually beneficial results and good agreements. The focus is on strategy, because the agreement on requirements regarding the interoperations is political first, then technical, and implies the partners sharing the same long-term vision.

Constructive Interoperability is related to engineering processes, standards, and specific technical agreements to be put in place in order to achieve interoperability. Thus, constructive interoperability is aligned with how interoperable systems are built. Currently available sets of standards are nevertheless not sufficient today, as they are not covering all the required aspects, as for example data quality. Recommended practices and certification for conformance are most of the time to be defined in order to ensure that data produced by a provider are those expected by the consumer.

Operational Interoperability is related to the ability of interoperating systems to contribute to achievement of a greater human goal in the operational environment. It implies delivering the right information at the right time and in the right format, but also consistently delivering it without undue effort on the part of humans, including after system upgrades. Operational interoperability is thus always the ultimate goal, and involves (for example) common methodologies and operational procedures as well as collaborating systems.

6.4.3 Federated Framework

The federated framework combines the three previously mentioned frameworks, and aims to resolve interoperability issues for Networked Collaborative Product Development in the extended enterprise. It aims to achieve pragmatic interoperability at an acceptable price, by federating the usage of interoperability enablers and ensuring to achieve identified brakes for interoperability. Both interoperability enablers and brakes are managed evolving lists capturing experience on the domain of interoperability. The goal is to be able to build an evolutionary framework, keeping advantage of results developed over years, and to put the focus on issues never addressed or resolved appropriately by research community. In addition, the federated framework gives a more precise focus on the systems considered for which interoperability is to be achieved: the organizations with processes supported by enterprise applications, which are realized by Information and Communication Technologies (ICT). Enterprise application is the cornerstone of the framework, which results of deployment and operation of instances of software products in operational organization and operational ICT infrastructures. Finally, federated framework extends the approach defined by ATHENA by providing a focus on:

- Importance of the Interoperability Maturity Level of an eBusiness community willing to achieve interoperability;
- Importance of addressing data loss coming from usage of different paradigms and technologies;
- Importance of relying on open standards and open source technologies for preparing the interoperability;
- Importance of Model Driven Architecture for building fast interoperability.

Today the federated interoperability framework usage is restricted to a community of experts, and different operational projects and research projects have been contributing to its robustness and applicability to support strategy of European aeronautic and space in terms of PLM standards (c.f. ASD SSG later in the case study section).

6.5 Case Studies

This section exposes two use cases, one in the aerospace industry and the other in the B&C domain. Both domains apply interoperability strategies: the one in the aerospace industry implements the federated framework exposed in the previous section and the second use case commits with domain standards.

6.5.1 In the Scope of European Aerospace and Defence

European Aerospace and Defence identified importance of standards in order to implement PLM approach for development of their products in the supply chain. It is illustrated by the activities of the ASD Strategic Standardization Group (SSG).

The SSG was set up in October 2008 by a group of European manufacturers, A&D associations and military governmental agency in order to share efforts of development of common A&D e-Business standards and associated harmonized European policies for operational use.

Both are understood today as two strategic levers of competitiveness for all the A&D manufactures and their supply chains (New role of the Original Equipment Manufacturer as System Integrator, and product and process Information Integrator).

The ASD SSG does not aim to create new eBusiness standards but to support effective governance at European level of International and European standards:

- Identifying a set of standards to use or to develop in order to cover the full spectrum of needs for eBusiness;
- Proposing and applying governance tools at strategic and technical level (e.g. radar screen, interoperability framework, assessment process);
- Developing a network of experts;
- Developing liaisons with all relevant standardization organizations.

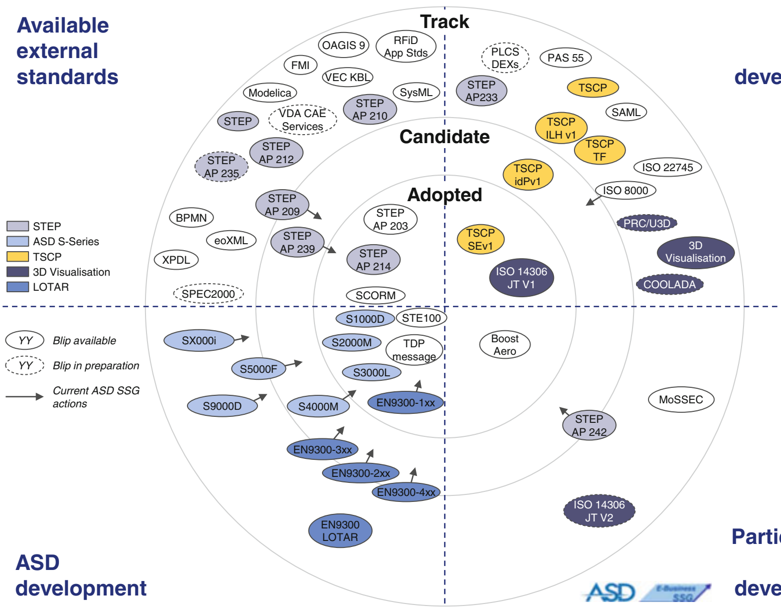


Fig. 6.5 ASD radar screen

Figure 6.5 shows the Radar screen of ASD SSG with tracked, candidate and adopted standards of the European Aerospace and Defence. Considered standards are ASD external standards (e.g. adopted STEP AP214 “Core data for automotive mechanical design processes”), standards for which ASD SSG monitor the development (e.g. JT), external standards for which ASD SSG participate to the development (e.g. STEP AP242) and standards ASD develops (e.g. TDP Message).

Principles of ASD SSG illustrate the concept of Interoperability Maturity Level of eBusiness Community, defined in the federated framework. This community is building a referential, which will allow preparing interoperability, as defined by ISO. The standards identified include ISO STEP standards. Various working groups of the ASD SSG are addressing important topics such as: PLM interoperability, Integrated Logistic Support, Hub collaboration, Supply Chain Interoperability, Data quality, Through Life Cycle interoperability. Figure 6.6 is the scope addressed by one sub topic of the PLM interoperability working group, the CAD 3D composite interoperability.

The topics are defined according to the priorities collected from the involved industrial companies and current challenges. Operational results have been achieved through PLM policies of some companies, Operational PLM Hubs (BoostAerospace, c.f. Fig. 6.7) or involvement for establishment of implementers’ forums, which is the current sensible topic for fast deployment, and usage of elected standards. At this stage, European Aerospace and Defence progressed well these last five years for reaching a good interoperability maturity level.

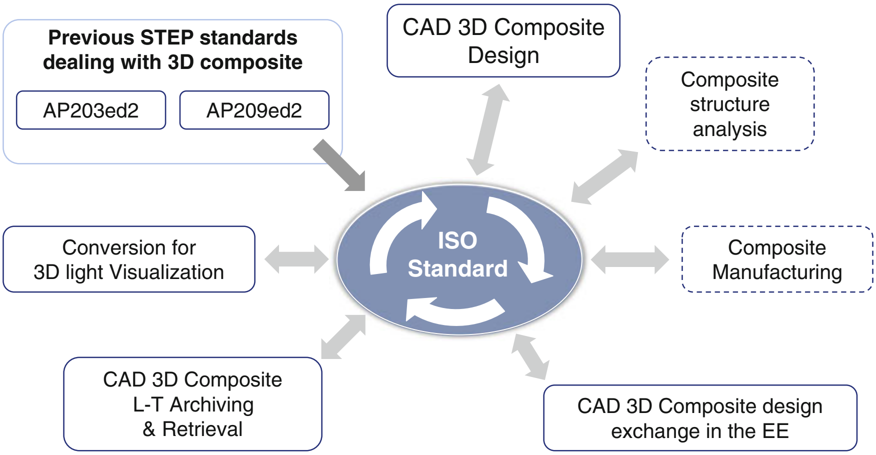


Fig. 6.6 Scope of CAD 3D composite interoperability by ASD SSG

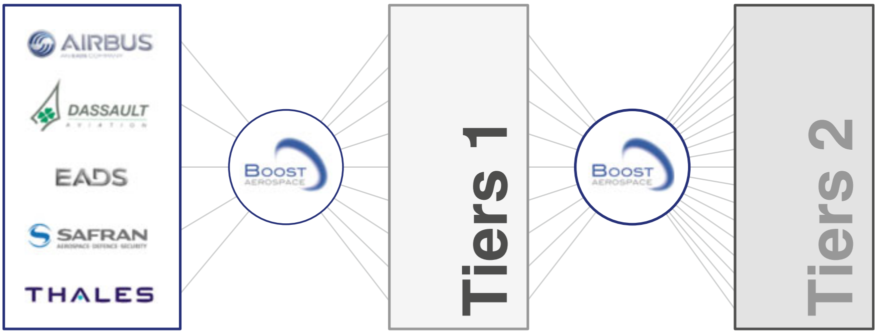


Fig. 6.7 BoostAerospace: PLM hub for aerospace supply chain

But the target remains challenging, and requires relying on research in order to improve flexibility for creation of Dynamic Manufacturing Networks. It is addressed for example by the IMAGINE project (c.f. [64]) for which an Aeronautic Living Lab extends the Federated Framework and provide an infrastructure which will then be able to support activities of the ASD SSG and the standardization community.

The Aerospace and Defense Living Lab objective is to support eBusiness collaboration for Networked Collaborative Product Development. Aerospace and Defense Living Lab demonstration is one of the five demonstrations of the IMAGINE results. For the considered Business Cases and scenarios, many enterprises working on the different parts of an aircraft are organized as a dynamic manufacturing network (c.f. Fig. 6.8), each member having a set of capabilities related to Computer Aided Engineering, Computer Aided Manufacturing and Computer Aided Collaboration. The DMN methodology will be used on top of one

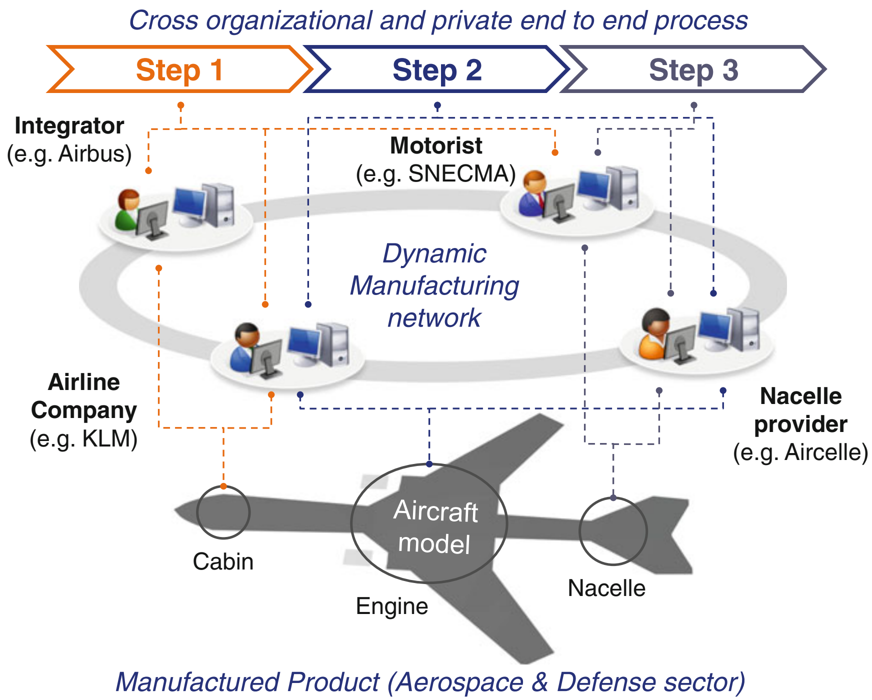


Fig. 6.8 Description of the targeted dynamic manufacturing network

implementation of the i_platform, which is a collaborative platform developed during IMAGINE project and aiming at dynamic definition, monitoring and adaptation of efficient end-to-end processes, combining cross organizational processes and private System Engineering and PLM processes. A strong focus will be on the "digital product" usage and secured interchange of product and process data within the extended enterprise, and between the technical enterprise applications or authoring tools supporting the processes.

Technical enterprise applications may be PDM systems, Enterprise Resource Planning System, Maintenance, Repair and Operations systems, Virtual Product Manager systems, etc.

Authoring tools may be Computer Aided Design tools or Simulation tools for a set of different disciplines and fields which are important for the considered systems, such as powerplant, aerodynamic, acoustic, etc.

Currently, the new trends are leading to a full reconfiguration of the Supply Chains within European Aeronautic and Defense, for various families of products. New projects are reaching a high level of subcontracting (e.g. 60 % for Airbus' A380) and are targeting even higher level of subcontracting for future programs (e.g. future long-range Aircrafts programs are targeting 80 %). In addition, it is targeted to reduce the number of tier-one sub-contractors, but as these are also sub-contracting a lot, there is a global increase of the number of sub-contractors for the whole supply chain from level one to the other levels. This trend can be referred as "Virtualization of the enterprise". In addition, another trend is the systematic usage of Computer and Modelling for a growing set of engineering disciplines and for collaboration, leading to systematic usage of Model-Based engineering for System Engineering or Product Life Cycle Management. This trend can be referred as "Virtualization of the Product". The combination of the both trends is leading to border effects in terms of interoperability for collaboration within extended enterprise implying interchange and sharing of digital model of the product. In order to respond to such needs, emerging eBusiness PLM Hubs within large groups (EADS PHC PHUSION) and for the European Aerospace and Defense are being setups, with on the one hand the systematic usage of COTS as components of the collaborative infrastructure and digital engineering chains, and on the other hand the identification of strategic importance of eBusiness PLM standards, as promoted by SSG such as ASD Strategic Standardization Group or EADS Strategic Standardization Committee.

In such a context, some issues exist with the expected qualities of a cross organizational collaborative platform:

- Interoperability, flexibility, robustness, security or other qualities of the platforms are to be adaptive in order to support a continuously changing Supply Chain without endangering the programs. Current platforms don't yet support it.
- On boarding process, i.e. the process for an enterprise to enter the collaborative network and connecting his processes and applications, is facing heterogeneous maturity of the members of the Supply Chain when dealing with digital collaboration, making it difficult to constitute Dynamic Manufacturing Networks;
- The appropriate methodologies for setting-up effective and adaptive end-to-end processes combining internal private processes and cross-organizational collaborative processes through such a platform don't exist yet;
- Organizational impacts of the strategy adopted by the Hub initiatives in terms of Architecture, Security and PDM are not always well assessed by involved organizations and stakeholders because of the complexity of these new environments and because of the lack of experience in deploying proposed approaches at this scale;
- PLM strategic organizations are facing some difficulties for making a Design Office, Production, Customer support and Supply Chain Management communities working together and combining their effort for end-to-end processes along all phases of the product life cycle.

The Aerospace and Defense IMAGINE Living Lab is an experimental environment with an applicative infrastructure, dedicated to the different addressed business cases, based on open standards and enacted on the Cloud (i.e. made available as services on the web and hosted outside of enterprise). It includes:

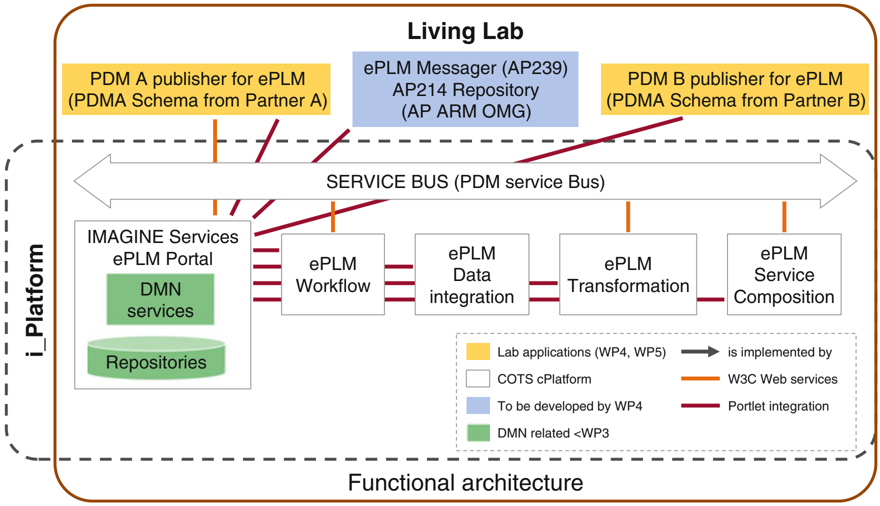


Fig. 6.9 Applicative architecture of aeronautic LL with inclusion of i_platform

- An implementation of the i_platform based on Open Source Components;
- A set of applications simulating actual enterprise applications that will be connected to the i_platform in order to constitute the network of applications required for the support of the Dynamic Manufacturing Network. Figure 6.9 gives an example with PDM systems of DMN members, and PLM collaborative platform constituting a PLM Hub (cPLM);
- A physical machines infrastructure allowing to simulate interconnected intranets and extranets;
- A set of complementary services and components required to ensure interchange of data between partners, processes and applications (e.g. cPLM Data integration component and cPLM transformation component in Fig. 6.9);
- A set of complementary components for ensuring appropriate service composition and quality of service for the applicative infrastructure (e.g. cPLM Workflow and cPLM Service Composition in Fig. 6.9).

The IMAGINE Living Lab is an experimental environment implementing a set of standardized business processes based on open standards, which are:

- The ISO 10303 standards for manufacturing data exchange, sharing and long term archiving;
- The ISO 15288 standards for system engineering, providing a process framework for development of a manufactured system and associated supporting systems, such as design or production capabilities;
- The ISA95 standard, for Business Production integration;
- The Supply Chain standards such as SCOR or BoostAero.

The IMAGINE relies on associated recommended practices for setting up and operate a Networked Collaborative Product Development platform, including the DMN methodology adapted to the Aeronautic Living Lab context. The Network Collaborative Product Development environment processes comprise the following parts:

- P1—Set up of the platform (Design, development, deployment and enactment);
- P2—Interconnection between Information system of a company and the collaborative platform;
- P3—Collaborating (several instances of collaboration processes);
- P4—Leaving the community (disconnection from the Collaborating platform).

The Aeronautic Living Lab architecture has been designed to solve the existing problems identified previously. An architecture framework based on open standards is used for ensuring interoperability, flexibility and security. Recommended practices and DMN methodology aim to increase Manufacturing community's maturity for practices based on new solutions developed by IMAGINE. The Aeronautic LL Business Cases will combine the standards of Production as well as the ones for Design, Supply Chain Management, Integrated Logistic and Support and System Engineering.

Doing so, experts in charge of defining strategic governance of eBusiness PLM standards have new means that should make it possible to implement usage of standards in Dynamic Manufacturing Networks.

6.5.2 In the Scope of Building and Construction

The B&C domain is one of the biggest industries in the world. Nevertheless, the organization of this industrial sector is very complex and fragmented, where over 95 % of the companies are small and medium-sized enterprises, nationally or geographically oriented.

In today's global and collaborative society, the success of business relies on the ability to collaborate. Better information and communication between all parties involved allows a reduction of the time required to develop a new product. Products can be delivered earlier and more professionally, thus permitting earlier and better returns on investment.

In spite of the emergence of ICT and their successful application by various manufacturing industries, communication in B&C has a low success rate, due to its fragmented sector. Although B&C has dynamic partnerships between various experts from different organizations, the reality is that each partner uses isolated systems that are mostly not interoperable and that cannot support data interface standards. Manual transfer of data is still largely the way to pass information. Tasks are therefore undertaken as isolated notes of a large network, in which communication procedures are primitive, slow and expensive. Interoperability is a prime factor for better control of quality and time and for cost reduction.

The B&C industry involves a long process directed at the satisfaction of the needs of clients and customers through the provision of quality products that fulfill their purpose at reasonable cost. It comprises complex activities that involve the combined efforts of several specialists from different disciplines.

Contrary to what is seen in other manufacturing industries, the B&Cs artifact is almost always a one-off product. This main difference between the creation of a single item and mass production has led to the adoption of a conservative technological attitude, while progress has not been as fast as in other industries. The life cycle of an artifact is long and normally it involves a large number of participants with different experience and knowledge, often located in different geographic areas.

In B&C the design and development of a product is basically a sequential process comprising various stages in which the following stage does not begin before the previous one is concluded. Nevertheless, large projects allow concurrency between the different stages. For example, construction begins before the design and planning stage is complete, allowing a reduction of the overall project time, while phased occupation begins before final project completion, providing an earlier return on investment. The major obstacles identified to block such an approach are caused by two main facts: (1) engineering data is not interoperable; (2) interaction between different participants is neither represented nor correlated.

The need of a unified and interoperable model that integrates all information and knowledge related to the different stages and that allows participants to access all information is a requirement. In this scenario, the end user can directly access the system's data or do it through the project manager, controlled by the system's managing applications and rules (Fig. 6.10).

The main benefit is to have all information interoperable using a neutral, accessible and exchangeable format, which is faster and much less prone to error. This avoids re-entering of information that is already in one of the components of the system and stimulates reuse and the share of information, since, once introduced in any of the components of the system, will be kept and available in the global system.

Nowadays, several standards are already available and other models are in advanced stages of development. It is expected that further proposals for the development of new models will come out due to the continuous need for the applications to extend their industrial scope.

The development of standard-based product and process data models to support the life cycle activities for the B&C industry has been addressed intensively during the last decade. In Europe, for example, several projects have worked on this issue, proposing and developing frameworks to assist in the establishment of interoperable open environments for the B&C industry.

Examples of major projects of the early 90s that were the first to research and develop standard-based methodologies and integrated platforms for data exchange, sharing and achieving in B&C domains, can be listed as follows: (1) the COMBINE project (European JOULE program), developed a computer-based integrated building design system and demonstrated the feasibility of technology proposed by

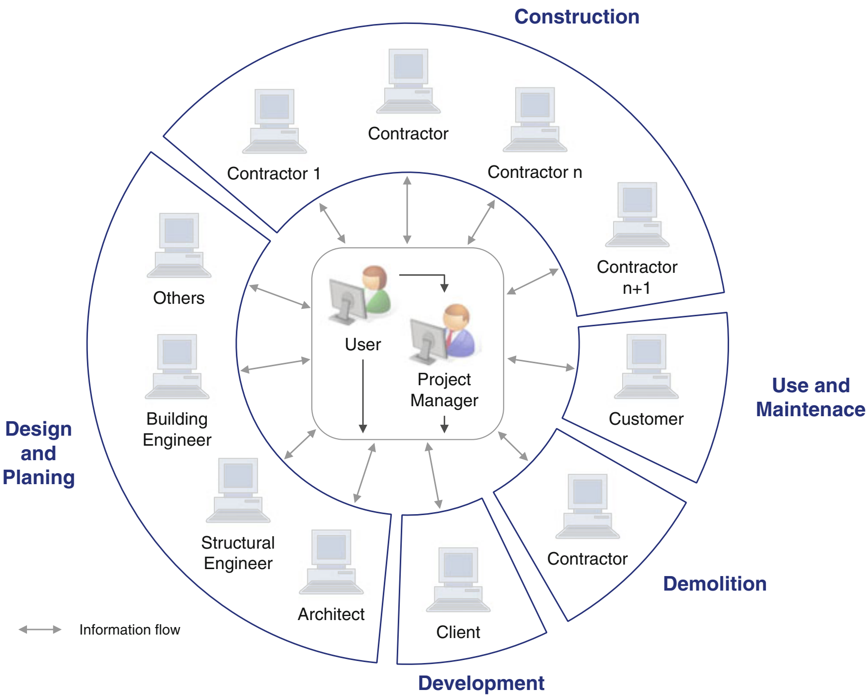


Fig. 6.10 Integrated project model

STEP to integrate applications within the building field through a common integrated data model; (2) the ATLAS project (European ESPRIT III program), defined a methodological framework and software tools conforming to STEP to enable the integration of large scale engineering applications;

During the second half of the 90s, the VEGA project (ESPRIT IV program) presented a proposal to bridge the gap between four standards (STEP, SGML, CORBA, EDI) developing the COrba Access to STep models (COAST) platform and services for distributing CAD applications.

Also, the project "C-ECOM: Cluster for Electronic Commerce" was an active concerted network sustaining a group of projects working on standardization activities in the frame of research and technological development in the area of "New methods of work and e-Commerce". C-ECOM is showing that standards are necessary to enable organizations to trade e-globally.

Regarding the utilization of state-of-the-art IT in the building industry, adherence is still very poor. Most established developments are concentrated on the Computer-Aided Design (CAD), drafting, scheduling control of tasks, and automation of certain pre-fabrication processes. Each application very often runs alone, without the capability to exchange the necessary data automatically. This situation has driven companies to reach proprietary dependent departmental solutions, without a complete integration of product, process and business data.

For instance, nowadays there is still little attention given to the automated data flow of planning and control information, or promoting the business need to the seamless flow of scheduling, resource, materials, and cost information between firms. For practitioners in these industrial environments, time and material planning is very often done in a manual process at the construction site on a stand-alone basis, mostly assisted by data received by phone calls or on paper.

The application of a Computer Integrated Environment including design, planning, production, business and control to an outdoor building site can then be seen as a top stage of IT integration into the construction industry.

The adoption of effective exchange of information regarding planning and control activities (schedules, resources, materials, cost, cash flow) between the different parties involved in B&C projects is a critical success factor. It may avoid projects' time and costs overrun and assure better quality.

The main interest of future work lies in the development and elaboration of a generic architecture for standard-based integrated indoor and outdoor construction sites. Therefore, further work should also be concentrated on the automation of specific heavy-duty machines in order to obtain a manpower-reduced building site, with computer-integrated time scheduling and material-planning possibilities. Integration with commercial aspects related with the B&C business should also be considered when aiming for a complete and automated data flow among internal (construction) and external (business) actors performing in this industrial sector.

The adoption of standards in modeling using a specific data protocol to support the integration and flow of information using a unique data exchange format, to be adopted by the various activities in B&C environments is a key factor for proposals now coming out. Examples are the work being developed by STEP within ISO TC184/SC4 B&C Working Group, by International Alliance for Interoperability throughout its Industrial Foundation Classes (IAI/IFC), and UN/EDIFACT (United Nations Electronic Data Interchange for Administration, Commerce and Transport) and its related adaptations.

6.5.2.1 Industry Foundation Classes Model Architecture

The Industry Foundation Classes (IFC) model architecture has been developed using a set of principles governing its organization and structure. These principles focus on basic requirements and provide a structure for the model together with a framework for sharing information between different disciplines within the AEC/FM industry.

This architecture provides a structure identified as the 'model schemata', and it has four conceptual layers within the architecture, which use a strict referencing principle. Within each conceptual layer a set of model schemata are defined.

The first conceptual layer is the *resource layer*. It provides resource classes used by classes in the higher levels. The second conceptual layer, i.e., the *core layer*, provides a core project model. This core contains the kernel and several core extensions. The *interoperability layer* is the third conceptual layer. It provides a set of modules defining common concepts and objects across multiple application types and AEC industry domains. Finally, the fourth and highest layer is the *domain layer*. It provides a set of modules tailored for specific AEC industry domain or application type. There are three possible ways to share data using IFCs. These are:

- By creating a physical file of information that may be shared across a network, by email or on a physical medium such a floppy disk. The EXPRESS language specification view of the IFC Object Model determines the structure of the file and the syntax of the file is determined by ISO 10303 part 21, or more recently in XML;
- By placing information in a database that has an interface defined according to the ISO 10303 part 22 (Standard Data Access Interface) for putting in and getting out data. The EXPRESS language specification view of the IFC Object Model determines the structure of the information sent to or received from the database. Presently, a number of software applications work using shared databases (also known as project model servers);
- By using software interfaces that can expose the information content of defined groups of attributes within an object. Software interfaces allow for direct communication between applications without the need for an intermediate file or database.

6.6 Conclusions and Outlook

In this chapter we have presented standard languages, data models, methods to support interoperability, a federation of frameworks for resolving interoperability and two case studies showing that interoperability is achieved or can be achieved through standardization.

In the future, servitization of manufacturing industries, i.e. the innovation of organization's capabilities and processes to shift from selling products to selling integrated products and services that deliver added value, will lead. This has already started, as illustrated by initiatives such as BoostAerospace, or role of Cloud solutions for setting up infrastructure of IMAGINE project or Standard Interoperability PLM. Indeed, servitization provides means for companies to move up the value chain and exploit higher value business activities. A service-led competitive strategy sees everything-as-a-service where Cloud Computing is seen as a major trend. In the Cloud paradigm, interoperability is achieved by the standard specification of Application Programming Interfaces for business services enabling a service to interact with others [65, 66]. These drives the current efforts of the ASD SSG for pushing merging AP203 and AP214 into AP242 in order to ensure common standardized interfaces will be adopted by automotive and aerospace communities, but also other industrial sectors relying on ISO 10303 STEP standards, like building or furniture. It is expected it will address previous failures in attempting to defined standardized PDM or PLM services, by initiatives such as

OMG PDM Enablers [67], PDT.net [68], OMG PLM services [69] or OASIS PLCS Web Services [70].

Services will be ubiquitously available, which enhances Extended Enterprise collaboration. Servitization opens routes for new challenges such as product-as-a-service design, where semantics will be taken into account from the start with Linked Services and Linked USDL approaches.

Another trend is the use of agile design and development, where fast prototyping, test-driven, model-driven and behavior-driven development methodologies allow focusing on business cases and semantic interoperability issues.

Finally, being able to test at the earlier stage standards and their implementation, being software tools or business processes, will required new approaches in order to establish interoperability at an acceptable price [71].

Establishment is required of an implementer forum hosted by neutral organization, and based on new methods and approaches for testing cross-organizational collaborative processes within the extended enterprise, and all along the supply chain.

The tests to be performed will not concern one single standard, but a consistent configured set of complementary standards, dealing with data interchange as well with security or flexibility.

Industrialization practices will have to be qualified and continuously improved in order to achieve sustainable and continuous interoperability, despite continuous evolution of the ICT which is imposed by providers. What will be the next ICT trends after Cloud computing, Linked Services and Big Data and HTML5?

References

1. Ford C, Colombi J, Graham S, Jacques D (2008) A survey on interoperability measurement. In: 12th ICCRTS conference adapting C2 to the 21st century, Newport, 19–21 June 2007, CCRP. http://www.dodccrp.org/events/12th_ICCRTS/CD/html/papers/096.pdf. Accessed 15 May 2014
2. Morris EJ, Levine L, Place PR, Plakosh D, Meyers BC (2004) System of system interoperability. Software Engineering Institute. http://resources.sei.cmu.edu/asset_files/TechnicalReport/2004_005_001_14375.pdf. Accessed 15 May 2014
3. ISO 10303, STEP, AP214, Conformance class 8—recommended practices for the usage of STEP AP214 CC8
4. ISC, International STEP centers. Available at http://www.pdesinc.org/
5. Raman D (1999) XML/EDI cyber assisted business in practice, TIE Holding NV
6. Berre A (2002) IDEAS project report, Introduction to ebXML and web services
7. Wang L, Shen W, Xie H, Neelamkavil J, Pardasani A (2002) Collaborative conceptual design—state of the art and future trends. Comput Aided Des 34:981–996
8. Bray T, Paoli J, Sperberg-McQueen CM, Maler E, Yergeau F (2006) Extensible markup language (XML) 1.0, 4th edn, W3C recommendation 16 Aug 2006
9. Christensen E, Curbera F, Meredith G, Weerawarana S (2001) Web services description language (WSDL) 1.1. W3C Note, 15 Mar 2001
10. Chinnici R, Moreau J J, Ryman A (2007), Web services description language (WSDL) version 2.0 part 1: core language. W3C recommendation, 26 June 2007

11. Haller A, Gomez JM, Bussler C (2005) Exposing semantic web services principles in SOA to solve EAI scenarios. In: WWW 2005 conference, Chiba, Japan
12. Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5 (2):199–220
13. Beckett D (2004) RDF/XML syntax specification (revised). W3C recommendation, 10 Feb 2004
14. Brickley D, Guha RV (2004) RDF vocabulary description language 1.0: RDF schema. W3C recommendation, 10 Feb 2004
15. McGuinness DL, Van Harmelen F (2004) OWL web ontology language overview. W3C Recommendation, 10 Feb 2004
16. Horrocks I, Parsia B, Sattler U (2009) OWL 2 web ontology language: direct semantics. W3C working draft, 21 Apr 2009
17. Baader F, Calvanese D, McGuinness D (2003) The description logic handbook; theory, implementation, and applications. Cambridge University Press, Cambridge, MA
18. Martin D, Burstein M, Hobbs J, Lassila O, McDermott D, McIlraith S, Narayanan S, Paolucci M, Parsia B, Payne T, Sirin E, Srinivasan N, Sycara K (2004) OWL-S: semantic markup for web services. W3C Member Submission, 22 Nov 2004
19. Martin D, Hodgson R, Horrocks I, Yendluri P (2006) OWL 1.1 web ontology language. Submission Request to W3C
20. De Bruijn J, Keller U, Kifer M, Lausen H, Krummenacher R, Polleres A, Predoiu L (2005) Web service modeling language (WSML). W3C member submission, 3 June 2005
21. Farrell J, Lausen H (2007) Semantic annotations for WSDL and XML schema
22. SAWSDL W3C Committee, 2007, SAWSDL (Semantic annotations for WSDL and XML schema). Available at http://www.w3.org/TR/2007/REC-sawsdl-20070828/#Example
23. Verma K, Li K, Brewer D (2005) Lumina—semantic web service discovery. Available at http://lsdis.cs.uga.edu/projects/meteor-s/downloads/Lumina
24. METEOR-S, (2007) METEOR-S download and release page. Radiant: WSDL-S/SAWSDL annotation tool. Available at http://lsdis.cs.uga.edu/projects/meteor-s/downloads/index.php?page=1
25. Patil AA, Oundhakar SA, Sheth AP, Verma K, Kunal V (2004) Meteor-S web service annotation framework. In: Proceedings of the 13th conference on World Wide Web WWW 04. ACM Press, New York, p 553
26. Sheth AP, Gomadam K, Ranabahu A (2008) Semantics enhanced services: METEOR-S, SAWSDL and SA-REST. IEEE Data Eng Bull 31(3):8–12
27. Lausen H, De Bruijn J, Polleres A, Fensel D (2005) WSML—a language framework for semantic web services. In: Proceedings of the W3C workshop on rule languages for interoperability, Washington DC. http://www.w3.org/2004/12/rules-ws/paper/44/, Accessed 15 May 2014
28. Klusch M (2008) Semantic web service coordination. Intell Serv Coord Semant Web pp 59–104
29. Haller A, Gomez J M, Bussler C (2005) Exposing semantic web service principles in SOA to solve EAI scenarios. In: WWW 2005 conference, Chiba, Japan
30. Zaremba M, Oren E (2005) WSMX execution semantics, WSMX working draft D13.2 v0.2
31. Dimitrov M, Simov A, Momtchev V, Konstantinov M (2007) WSMO studio—a semantic web services modelling environment for WSMO (system description). In: 4th European semantic web conference, vol 4519. Springer, Berlin, pp 749–758
32. Super Project (2009) Semantic web services
33. Klusch M (2008) Semantic web service description. Intell Serv Coord Semant Web pp 31–57
34. Burstein M, Ankolenkar A, Paolucci M (2003) DAML-S: semantic markup for web services, The DAML services coalition
35. Knublauch H, Hendler J A, Idehen K (2011) SPIN—overview and motivation. Available at http://www.w3.org/Submission/spin-overview
36. Scicluna J, Abela C, Montebello M (2004) Visual modelling of OWL-S services. In: IADIS international conference WWW/Internet, Madrid, Spain

37. Giampapa J, Paolucci M, Srinivasan N, Vaculin R, Group S (2008) OWL-S 1.1 API. Available at http://projects.semwebcentral.org/projects/owl-s-api
38. Paolucci M, Kawamura T, Payne T, Sycara K (2002) Semantic matching of web services capabilities, In: The semantic web—ISWC 2002, vol 2342/2002, pp 333–347
39. Jaeger M C, Rojec-Goldmann G, Liebetruth C, Mühl G, Geihs K (2005) Ranked matching for service descriptions using OWL-S. In: Kommunikation in Verteilten Systemen 2005 (KiVS 2005), pp 91–102
40. Klusch M, Fries B, Sycara K (2009) OWLS-MX: a hybrid semantic web service matchmaker for OWL-S services. Web Semant Sci Serv 7(2):121–133
41. Klusch M, Fries B (2008) Hybrid OWL-S service retrieval with OWLS-MX: benefits and pitfalls, In: di Noia T et al. (eds) Proceedings of the workshop on service matchmaking and resource retrieval in the semantic web (SMR2-2007), Busan, South Korea, 11 Nov 2007, p 15. http://ceur-ws.org/Vol-243/. Accessed 15 May 2014
42. Cabral L, Domingue J (2009) Translating semantic web service based business process models. In: 2009 IEEE Asia Pacific services computing conference APSCC, pp 1–6
43. Cardoso J, Barros A, May N, Kylau U (2010) Towards a unified service description language for the internet of services: requirements and first developments, vol 24 IEEE, pp 602–609
44. Pedrinaci C, Domingue J (2010) Toward the next wave of services: linked services for the web of data. J Univers Comput Sci 16(3):1694–1719
45. Pedrinaci C, Cardoso J, Leidig T (2014) Linked USDL: a vocabulary for web-scale service trading. In: 11th extended semantic web conference (ESWC 2014). Springer, Berlin, http://people.kmi.open.ac.uk/carlos/wp-content/uploads/downloads/2014/04/linked-usdl.pdf. Accessed 15 July 2014
46. Josuttis NM (2007) SOA in practice, vol 253. O'Reilly, Sebastopol, p 352
47. ISO/IEC JTC-1 (ISO) (1993) ISO/IEC-2382-01: information technology—vocabulary—part 1: fundamental terms
48. IDABC (2004) European interoperability framework for pan-european egovernment services
49. Fensel D, Bussler C (2002) The web service modeling framework WSMF. Electron Commer Res Appl 1:113–137
50. Bouras A, Gouvas P, Kourtesis D, Mentzas G (2007) Semantic integration of business applications across collaborative value networks. In: Camarinha-Matos LM et al (eds) Establishing the foundation of collaborative networks. Springer, New York, pp 539–546
51. Izza S, Vincent L, Burlat P (2006) A framework for semantic enterprise integration. In: Interoperability of enterprise software and applications. vol 5, section 2. Springer, London, pp 75–86
52. Ferreira da Silva C, Cunha PR, Ghodous P, Melo P (2010) The semantic side of service-oriented architectures. In: Mentzas G, Friesen A (eds) Semantic enterprise application integration for business processes service oriented frameworks. IGI Global, Hershey, New York, pp 90–104
53. Kalfoglou Y, Schorlemmer M (2003) Ontology mapping: the state of the art. Knowl Eng Rev J 18(1):1–31
54. Bouquet P, Serafini L, Zanobini S, Sceffer S (2006) Bootstrapping semantics on the web. In: Proceedings of the 15th international conference on World Wide Web—WWW'06. ACM, New York, pp 505
55. Klusch M, Fries B, Sycara K (2006) Automated semantic web service discovery with OWLS-MX. In: Proceedings of the 5th international joint conference on autonomous agents and multi-agent systems. ACM New York, pp 915–922
56. Kiefer C, Bernstein A (2008) The creation and evaluation of iSPARQL strategies for matchmaking. In: Bechhofer S et al (eds) Proceedings of the 5th European semantic web conference on the semantic web: research and applications. Springer, Berlin, pp 463–477
57. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y (2007) Pellet: a practical OWL-DL reasoner. Web Semant 5(2):51–53
58. Li L, Horrocks I (2004) A software framework for matchmaking based on semantic web technology. Int J Electron Commer 8(4):39–60

59. Klusch M, Kapahnke P (2012) The iSeM matchmaker: a flexible approach for adaptive hybrid semantic service selection, web semant. Sci Serv Agents World Wide Web 15:1–14
60. Klusch M (2012) Overview of the S3 contest: performance evaluation of semantic service matchmakers. In: Blake B et al (eds) Semantic web services SE—2. Springer, Berlin, pp 17–34
61. Naeem M, Heckel R, Orejas F, Hermann F (2010) Incremental service composition based on partial matching of visual contracts, In: Rosenblum DS, Taenzer G (eds) Fundamental approaches to software engineering. Proceedings of the 13th international conference FASE 2010. Springer, Berlin, pp 123–138
62. Anastasiou M, Berre A-J, Elvesæter B, Figay N, Garcia O, Greiner U (2010) ATHENA interoperability framework AIF, European integrated project, 2010. Available at http://athena.modelbased.net
63. Figay N (2009) Interoperability of technical enterprise applications, Thesis report. http://www.eads-iw.net/web/plm-interop. Accessed 10 July 2014
64. NN (2011) IMAGINE Research & Development project. http://www.imagine-futurefactory.eu. Accessed 10 July 2014
65. Benfenatki H, Saouli H, Benharkat N, Ghodous P, Kazar O, Amghar Y (2013) Cloud automatic software development. In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 40–49
66. Benfenatki H, Kemp G, Ferreira Da Silva C, Benharkat AN, Ghodous P (2014) Service-oriented architecture for cloud application development. In: Cha J et al. (eds.) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 307–316
67. Product Data Management Enablers v1.3, Object Management Group, November 2000. Available at http://www.omg.org/spec/PDME
68. PDTnet Project—Product data technology and communication in an OEM and supplier network, ProSTEP Ivip, 2000. Available at http://www.prostep.org/en/alte-seiten/standard-info/pdt-net-projekt.html
69. Product Lifecycle Management Services, v2.1, Object Management Group, May 2011. Available at http://www.omg.org/spec/PLM/2.1/
70. PLCS Web Service v2, VIVACE, 2005. Available at http://www.plcs-resources.org/plcs_ws/v2/
71. Ottino A, Ghodous P, Ladjal H, Shariat B, Figay N (2014) Interoperability of simulation applications for dynamic network enterprises based on cloud computing—aeronautics application. In: Cha J et al. (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 597–606

Chapter 7
Collaborative Engineering

Milton Borsato and Margherita Peruzzini

Abstract Collaborative Engineering is the practical application of collaboration sciences to the engineering domain. Its aim is to enable engineers and engineering companies to work more effectively with all stakeholders in achieving rational agreements and performing collaborative actions across various cultural, disciplinary, geographic and temporal boundaries. It has been widely applied to product design, manufacturing, construction, enterprise-level collaboration and supply chain management. The present chapter clarifies the main concepts around Collaborative Engineering, as well as the various forms of collaborative ventures, such as virtual enterprises. It underlies the crucial impact of Collaborative Engineering in the context of global distributed engineering. The most applied forms of technology for collaboration are presented, such as Computer Supported Collaborative Design (CSCD) and web-based design, which are mature fields of study in constant improvement, as collaborative tools and cloud-based systems become more pervasive. The application of Collaborative Engineering in the context of product lifecycle is also discussed, and different needs for collaboration are evidenced along successive steppingstones of product development. Two case studies are provided to illustrate successful application of the concepts hereby provided.

Keywords Collaborative engineering · Collaboration · Product lifecycle · Virtual enterprise

---

M. Borsato (✉)
Federal University of Technology, Parana, Brazil
e-mail: borsato@utfpr.edu.br

M. Peruzzini
Polytechnic University of Marche, Ancona, Italy
e-mail: m.peruzzini@univpm.it

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_7

7.1 Introduction

One of the most significant characteristics of humankind is that we are predisposed to associate with peers to create groups. We are a social species, from the Latin word *socii* meaning allies [1]. These alliances are formed from the need of a social life. Thus, every human "*…needs others to satisfy his own needs… and he [sic] improves himself (becomes 'more human', exercises and develops his capacities) in his relationships with others*" [2].

Nowadays, in spite of this social bias, in many ideologies of the world people have a strong tendency towards individualistic behaviours that seek to accumulate resources and power for personal gain. Competing and selfish individualistic mental models reinforce a mutual process by which individuals tend to shape the culture of the group to which they belong. The result of such loop is that human greed and thirst for money and power may yield a perceived maximum gain for the individual, at least in the short-term, but it results in sub-optimal gains in the short- and long-term for the group [3].

In a long-term perspective, human collaboration is required. Collaboration, derived from the Latin word *collaborare* meaning **to work together**, is the process of multiple people working together interdependently to achieve a greater goal that is impossible for any individual to accomplish alone [4]. In a fully connected society, collaboration is ubiquitous in all professional activities, ranging from technical projects for business pursuits to international efforts, such as space exploration and combating global warming.

Collaborative Engineering is the practical application of collaboration sciences to the engineering domain [5]. The aim of this emerging human-centred discipline is to enable engineers and engineering companies to work more effectively with all stakeholders in achieving rational agreements and performing collaborative actions across various cultural, disciplinary, geographic, and temporal boundaries. Because products have become more complex, competitiveness has harnessed efforts in companies worldwide, and sustainability issues have raised concerns in society, collaboration is increasingly needed from inception throughout the disposal of a product. According to Shen et al. [6], Collaborative Engineering is a "*concept of optimizing engineering processes with objectives for better product quality, shorter lead-time, more competitive cost and higher customer satisfaction*". It has been widely applied to product design, manufacturing, construction, enterprise-level collaboration and supply chain management.

Recently such an approach is particularly emerging due to the modern challenges in engineering and design: market globalisation, short delivery times, rapid evolution of customer requirements, and complex industrial chain creation. As a consequence, new flexible organizations arise to satisfy the design and manufacturing needs and face the novel industrial scenario. Indeed, working groups must be organised in a new way to cooperatively act in distributed teams and allow a dynamic team configuration according to the specific objectives. At the same time, traditional tools may not properly support such collaborative teamwork in distributed spaces and new technologies must be adopted.

The present chapter aims to present the basic concepts behind Collaborative Engineering in the context of Product Development, for helping one reason whether the benefits of collaboration may overweight burdens such as lack of control and other inherent risks. The following sections categorize collaboration ventures, such as virtual enterprises and extended enterprises, which have been attracting attention since the 90's. Furthermore, current developments in technology to support collaboration are explored, as well as future trends.

7.2 Fundamental Concepts

In this section, the fundamental concepts of Collaborative Engineering are covered. Firstly, the distinctions between similar terms such as coordination, cooperation and collaboration are presented. Then, the types of collaborative ventures, such as virtual enterprises and crowdsourcing, are defined.

7.2.1 Coordination, Cooperation and Collaboration in Engineering

Gaver [7] proposes four major landmarks within the spectrum on how collaboration may take place. It is a **pervasive experience**, one of simply knowing who is around and something about what people are doing: that people are busy or free, meeting or alone, receptive to communication or not. Above it, there is a **general awareness**, which is necessary for all collaborative work, but the degree to which its focus is shared varies. An intense sharing of awareness characterizes **focussed collaboration**, when people work closely together towards a shared goal. Less is needed for division of labour, that common work practice in which a shared goal is divided and component tasks addressed separately. Finally, more casual awareness can lead to **serendipitous communication**, in which people realize the potential for productive work through chance encounters.

There are differences between three frequently misinterpreted terms: coordination, cooperation and collaboration. According to Lozano, **coordination** refers to activities performed by different individuals in order to make them compatible with a common purpose or result; **cooperation** refers to engaging in work on monitoring and evaluation, learning from each other and sharing experiences; and **collaboration** refers to using information to create something new, seeking divergent insights and spontaneity, jointly developing proposals, sharing information, planning joint workshops, and raising funds together among other activities. In other words, collaboration thrives on differences and dissent [3].

On the other hand, Mattessich et al. [8] define cooperation as "*characterized by informal relationships that exist without a commonly defined mission, structure or effort. Information is shared as needed and authority is retained by each organization so there is virtually no risk. Resources are separate as are rewards*". Coordination would be "*characterized by more formal relationships and understanding of compatible missions. Some planning and division of roles are required, and communication channels are established. Authority still rests with the individual organization, but there is some increased risk to all participants. Resources are available to participants and rewards are mutually acknowledged*". And collaboration "*connotes a more durable and pervasive relationship; (…) a full commitment to a common mission (…); Authority is determined by the collaborative structure. Risk is much greater. (…)*". The underlying mechanisms for coordination, as discussed by Grandori [9], varying from formal to highly informal mechanisms, are in not in the scope of this text.

In the context of this chapter, **coordination** is assumed to create a structured and shared information exchange by communication, information exchange, and adjustments of activities. **Cooperation** extends coordination and considers also resource sharing for achieving compatible goals. Finally **collaboration** is understood as a more demanding process in which entities share information, resources and responsibilities to jointly plan, implement, and evaluate a program of activities to achieve a common goal and therefore jointly generating value.

Similarly, Camarinha-Matos et al. [10] extend the list of similar terms, including networking and coordinated networking. In their view, **networking** basically involves communication and information exchange for mutual benefit, while **coordinated networking** involves aligning/altering activities so that more efficient results are achieved, in addition to communication and information exchange.

When collaboration addresses engineering topics, we deal with Collaborative Engineering. In particular, Concurrent Engineering is one specific approach to collaboration, which typically takes place between divisions within a single company [11]. For its intrinsic nature, geographically distributed teams whose members are spread across a country, continent or globally, it may also accomplish Concurrent Engineering, by creating coordinated networking.

Some misunderstanding can rise between Concurrent Engineering and Collaborative Engineering. Lu et al. [12] try to clarify this point by advocating that, while Concurrent Engineering addresses the sequencing issues of interacting separate decisions, Collaborative Engineering requires stakeholders to negotiate a single joint agreement, based on multiple decisions made by participating individuals. Concurrent Engineering is based on fundamentals such as multidisciplinary teams (people), tools and processes, just like the socio-technical model proposed by Toyota [13]. However it is Collaborative Engineering that emphasizes the search for consensus, the need for sharing responsibilities and the achievement of complicity in results.

7.2.2 Collaborative Ventures

The smallest element in societies is the individual, who, by allying and interacting with other individuals, creates or becomes part of groups, which, in turn, are part of or create organizations [3]. One way of classifying groups is by their maturity: (1) already established ones, or old groups, and (2) groups that will be created or new groups. Such classification is important since old groups have, to a large degree, established their interactions, routines and behaviours, while new groups in many cases are in a transition state where the interactions, routines and behaviour are being established. Whereas in an old group the members know each other's fashion of working and personal differences, in a new group the members need to spend time adjusting to one another. In many cases, manageable conflicts appear from personal differences. Once these have been resolved and each of the members knows how to deal with the others, the group is in its road to becoming an old group.

As globalization, technological innovation and market turbulence challenge traditional business logic, firms are experimenting with new organizational models [14]. Among these new models, the **virtual enterprise** or **virtual organization** is one of the most popular. Galbraith defines a virtual corporation as "*the exact opposite of the vertically integrated corporation*". Instead of covering all the activities a business comprises (i.e. from raw materials to the ultimate consumer) the virtual corporation contracts out for all activities except those in which it is superior [15]. As a result, a network of independent companies (i.e. each does what it does best) acts together as if it were virtually a single corporation.

There are three types of virtual enterprises (VEs): Short-term VE, Extended Enterprise and Consortium VE [16]. Short-term VEs would be set up for responding to specific market needs. A project would be split into several linked modules, to be addressed by each partner. Extended enterprises comprise supply networks, which are engaged in several projects over a more sustained period of time. Consortium VEs combine core competences to obtain work.

Thompson proposes a practical taxonomy of collaborative endeavours [17]. For him, Virtual Business Networks (or VBN) are companies coming together to co-operate to achieve some shared business goal by forming networks enabled by various forms of web-based technology. VBNs appear in many guises and names such as Collaborative Networks, Virtual Clusters, Virtual Enterprise Networks, Collaborative Supply Chains, Networked Enterprises and Star Alliances.

Thompson's classification yet suggests eight different kinds of VBN: Collaborative Supply Chain (CSC), Collaborative Supplier Network (CSN), Collaborative Product Development Network (CPDN), Enhanced Trade Association (ETA), Incubation and Acceleration Network (IAN), Subcontracting and Partnership Exchange (SPX), Technology-Led Ecosystem (TLE), and Virtual Enterprise Network (VEN).

Given the large diversity of manifestations of collaborative ventures, Camarinha-Matos et al. [10] propose another taxonomy of the various organizational forms, resulting in eleven categories. As far as product development is concerned, the most important categories are collaborative network, supply chain, virtual enterprise, virtual organization, extended enterprise and virtual team.

The first category, named collaborative network, is "*a network consisting of a variety of entities (e.g. organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, culture, social capital and goals, but that collaborate to better achieve common or compatible goals, thus jointly generating value, and whose interactions are supported by computer network*". The second category, named supply chain, is "*a stable long-term network of enterprises each having clear roles in the manufacturing value chain, covering all steps from initial product design and the procurement of raw materials, through production, shipping, distribution, and warehousing until a finished product is delivered to a customer*". The third category, named virtual enterprise, is "*a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks*". The fourth category, named virtual organization, is "*a concept similar to a virtual enterprise, comprising a set of (legally) independent organizations that share resources and skills to achieve its mission/goal, but that is not limited to an alliance of for profit enterprises*". The fifth category, named extended enterprise, is "*a concept typically applied to an organization in which a dominant enterprise 'ex-tends' its boundaries to all or some of its suppliers*". The sixth category, named virtual team, is "*similar to a virtual enterprise but formed by humans, not organizations*".

The paradigm of Mass Collaborative Product Development (MCPD) has gained popularity in the software domain with the creation of software like Linux and Apache. This model is now gaining popularity in the physical product and services domain in two forms: crowdsourcing and mass collaboration [12]. Crowdsourcing takes place when design and development are carried out in response to an open challenge with a reward, like InnoCentive, Quirky, Local Motors and Darpa's Vehicleforge program. In mass collaboration, a product emerges as a result of people with similar interests working together on an idea. Examples include the Arduino™ Controller and the Open Source Car (OScar). Products with a modular architecture are more suitable for the largely decentralized development and decision-making that is prevalent in the MCPD model [18].

7.3 Technology for Collaboration

In this section, the main technology trends behind Collaborative Engineering are presented. Computer Supported Collaborative Design is an area that enables multidisciplinary design to take place. Moreover, web-based design is emphasized as the pervasive means to share data in a connected world.

7.3.1 Computer Supported Collaborative Design (CSCD)

The application of Collaborative Engineering to product design is usually called Computer Supported Collaborative Design (CSCD) or Computer Supported Collaborative Work in Design (CSCW-D). CSCD is carried out not only among multidisciplinary product development teams within the same company, but also across the boundaries of companies and time zones, with increased numbers of customers and suppliers involved in the process [6].

CSCD is also referred as Cooperative Design, Concurrent Design, or Interdisciplinary Design. It encompasses the process of designing a product through collaboration among multidisciplinary product developers associated with the entire product lifecycle. This area of knowledge covers various lifecycle phases such as preliminary design, detailed design, manufacturing, assembly, testing, quality control, and product service as well as those from suppliers and customers.

Weiseth presents a typology of collaboration tools using a wheel metaphor, referred to as the Wheel of Collaboration Tools (WCT) [19]. Collaborative Working Environments (CWEs) should include, among others, these functionalities:

- a service-oriented approach to support interoperability and the capacity of composing services, giving as a result new composite services;
- an activity-oriented approach to support the automation of tasks that co-workers have to perform in order to fulfil certain activities in their work; and
- context-based collaboration and personalized and adapted interfaces, which enable co-workers to determine how the system should react, thus providing self-organization, self-adaptation and self-deployment.

CSCD has been profoundly related to other research fields, including Computer Supported Cooperative Work (CSCW) and Human–Computer Interaction (HCI). The term CSCW was first used by to describe the topic of an interdisciplinary workshop that was organized on how to support people in their work arrangements with computers [20]. The CSCW field emerged with the objective of studying the creation and impact of collaborative applications [21]. CSCW tools enhance the sharing experience of product data, bringing the full potential of real-time collaboration to all product design stakeholders, and numerous technologies or standard formats have come to maturity, such as PHP and XML for dynamic processing of web data, VRML or Java3D for 3D viewing within web applications, and STEP or JT for product information exchanges [22] (see Chap. 6).

Many people simply refer to CSCW by the term of Groupware, though others consider this to be too narrow. Generally speaking, the term Groupware is widely used in commercial software products, while CSCW is used more in the research community [6]. The most widely used CSCW techniques in collaborative design systems include groupware techniques for facilitating communication among design team members and context awareness techniques for enhancing coordination among team members.

HCI, on the other hand, had its initial R&D focus on interaction between one user and one computer. It was then extended to human–human interaction via networked computers, which is the essence of CSCD. Blackboard architecture [23], Distributed Artificial Intelligence (DAI) [24] and software agent technologies [25] have provided the foundation for developing collaborative design systems [26].

The Distributed and Integrated Collaborative Engineering Design (DICE) project, started in 1986 at MIT, made several novel contributions to collaborative design, with the introduction of synchronous and asynchronous communication tools, integration of qualitative and quantitative geometric reasoning, knowledge-based design, use of asynchronous teams for solving both symbolic and numeric constraints, design rationale capture, and collaborative negotiation. Agent-based design systems use a communicative and intelligent coupled network of problem solvers. A key requirement of these tools is the representation of knowledge in an appropriate form, based on the designers' needs, the context in which it is being used, and the nature of the information being conveyed [27].

With the globalization of the manufacturing industry, CSCD is required to support distributed design. Members on a collaborative team often work in parallel and independently using different engineering tools distributed at separate locations, even across enterprise boundaries and across various time zones around the world. The resulting design process is then called Distributed Collaborative Design. The objective of a design team has multiple facets, for example, optimizing the mechanical function of the product, minimizing the production or assembly costs, or ensuring that the product can be easily and economically serviced and maintained.

A successful implementation of CSCD requires new approaches, such as including an efficient communication strategy for a multidisciplinary group of people from the design and manufacturing departments to share and exchange ideas and comments; an integration strategy to link heterogeneous software tools in product design, analysis, simulation and manufacturing optimization to realize obstacle-free engineering information exchange and sharing; and, an interoperability strategy to manipulate downstream manufacturing applications as services to enable designers to evaluate manufacturability or assembleability as early as possible [28].

7.3.2 Web-Based Design

Support groups originally developed the Web for information sharing within internationally dispersed teams and the dissemination of information. It has initially become a convenient media to publish and share information relevant to the design process, from concept generation and prototyping to virtual manufacturing and product realization. Therefore, it has been adopted as the most popular implementation architecture of a collaborative product development (including design and manufacturing) tool.

Wikis, introduced in 1995 for writing documents collectively and further developed into the Wikipedia project in 2001, are databases of interactive web pages that allow members of a user group to collectively edit the same material from any computer with a web connection. A wiki is unique in that it closely emulates a real verbal discussion with the added feature of being persistent. Studies of wikis in the light of collaborative design show that wikis can support the activity, although the state of use and technology inhibit more efficient use. Wikis therefore provide a flexible and self-organizing platform that is especially useful from the point of view of early design, when the information and knowledge is unstructured, and from the point of view of collaborative design, where all communication is persistently recorded and loosely organized through user-defined tags [27].

Developments in information technology and the Web created fundamental shifts in the way engineering was done by enabling the use of many new tools for human interaction and collaboration. Dynamic business needs coupled with globalization required the product development teams (like design, manufacturing, sales and management) to do real-time sharing of product data in a concurrent manner (see Chap. 18).

Many researchers and working groups have been working hard to improve the current Web infrastructure and supporting tools. Web-based infrastructure has been used in a number of collaborative product design systems. In most cases, the Web is primarily used by multidisciplinary team members as a medium to share design data/information/knowledge; while in some cases, it is integrated with other related technologies and is used for product data management and project management.

However, web technology alone is not a complete solution to collaborative design systems, although it makes communication physically viable through a common network. In order to collaborate on a distributed design project, remote engineers and designers need active supports to coordinate their efforts. This coordination involves the translation of terminology among disciplines, locating/providing engineering analysis services, virtual prototyping services, and project management. Servers should not only be a repository of information but also provide intelligent services to help users to solve design problems [6].

In recent years, more and more collaborative design systems have been developed using Web Services as well as Semantic Web and Grid Computing techniques, showing evident evolution from the use of agent-based technology and active web servers.

As with many other systems initially developed for the desktop or closed network environments, CSCD is also currently moving towards the cloud. In other words, applying cloud computing in the context of collaborative design has produced the concept of product collaborative cloud design. It promises to make it possible to share enterprises' design resources through collaborative design systems, reuse design knowledge, by registering in a system where product design professionals can start collaborative design between different departments or different enterprises without building up their own collaborative design infrastructure to avoid repeated construction and investment [29, 30].

7.4 Collaboration for Product Lifecycle

In this section, Collaborative Engineering is presented in the context of the Product Lifecycle. Knowledge Formalisation is introduced, for depicting the different shapes of knowledge that are generated and converted along the path towards product realisation. The nature of each activity performed in the process requires different collaboration styles, which are presented next. Product models, which encapsulate knowledge in subsequent phases of the lifecycle, are then explored. Information management is thus depicted, as the flow of information and its control is crucial for successful product development projects.

7.4.1 Knowledge Formalisation

Product design is goal-oriented process generating a new product description, through a repeated cycle of reformulating the problem, synthesising a potential solution, evaluating it, and reformulating the problem again [31]. "*It integrates several specific domains and involves multi-disciplinary competences, each of which is performing their own task to converge on the same goal. (…) The process complexity usually requires managing team structure and roles, data evolution, decision-making activities, integration of different software solutions and knowledge formalization to pre-serve all generated information*" [32].

As single components can be individually developed, both individual and collaborative activities are carried out alternatively and require an intense information exchange. Consequently, design collaboration can be modelled as a continuous shift from individual to team dimensions, when human interaction is fundamental and data elaborated during individual work are shared, discussed and analysed [33]. Norman proposes a cognitive model of interaction to represent collaboration in the team dimension [34]. It consists of seven stages of action and involves the explicit modelling of exploratory and reactive behaviours. Such a model does not depend on the features of the collaborative environment but only on human interactive and cognitive processes.

In order to manage such complexity, design methods are elaborated. Among them, product design methods may be applied to all sorts of artifacts (and related services), ranging from simple component design to complex engineering systems (CES). CES comprises a number of components and processes with interdependencies during development. Their development involves decomposition followed by integration of the system. Typically, this would encompass system architecture development, then component development, and finally system integration [35].

Manufacturing CES demands a strong level of interaction and collaboration between design, marketing, planning, and an integrated supply chain to meet program cost, quality, and timing objectives. Thus, CES is the variety of product that can best benefit from the most recent advances in collaboration tools and methods. In the context of the present chapter, therefore, the phrase **product design** is to be associated with the design of CES.

A product's lifecycle comprises the time frame "*spanning from concept to end-of-life of a product*" [36]. It may be characterised by the following three phases: beginning-of-life (BOL) including design and production; middle-of-life (MOL) including use, service and maintenance; and end-of-life (EOL) featuring various scenarios such as reuse of the product with refurbishing, reuse of components with disassembly and refurbishing, material reclamation without disassembly, material reclamation with disassembly and, finally, disposal with or without incineration [37].

All phases in the lifecycle of a product are well suited for collaboration; however, due to its intrinsic nature of intensive exchange of knowledge and information, product development (or design) deserves closer attention. A product development campaign can be viewed as a technical activity with a human purpose (e.g. to achieve market success). Therefore, when a group of engineers, with competing lifecycle concerns, come together in a team to develop a new product, it becomes a complex socio-technical activity. Product design demands a wide variety of knowledge sources (i.e. heuristic, qualitative, quantitative, and so on). The design problem is then solved in a multi-stage, iterative, and collaborative process, with extensive communication and coordination among teams of experts in various disciplines [27].

It is challenging to manage the effective communication of product knowledge and appropriately represent it among different groups. More recently, Chandrasegaran et al. [27] propose a design process view of knowledge, which is concerned with the knowledge that is generated and used at various stages of the product development process. Figure 7.1 shows this interpretation of knowledge representation during a product lifecycle.

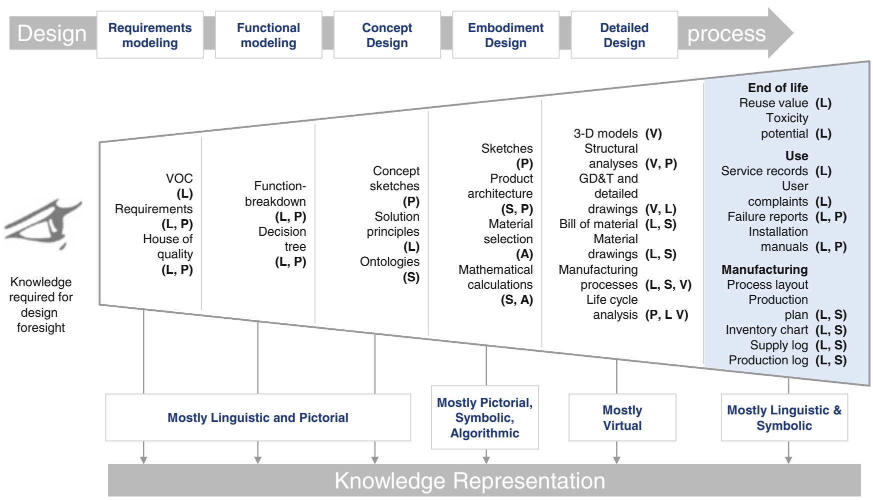


Fig. 7.1 Knowledge representations throughout the product lifecycle [27]

In the early stages of design, knowledge representation is predominantly linguistic and pictorial in nature. The other representations such as symbolic, virtual, and algorithmic appear in the embodiment design onwards, when much of the design is already committed. More information is accumulated as design reaches the embodiment and detailed design phases, and the challenge is to reuse or re-inject this knowledge into the earlier phases of design using an appropriate representation.

7.4.2 Collaborative Dimensions

Design collaboration assumes different characteristics according to the specific context, but there are some common features that can be identified in product design. Firstly, an important feature of collaboration in design is the need to integrate single pieces of work that are individually developed and must be integrated (i.e., tasks, decisions, analysis, product parts and subassemblies). Successful product design projects strongly rely on both the ability of the project leader to coordinate team participants and the mutual understanding of design viewpoints and decision sharing. Design process control is based on the knowledge of existing design situations, critical evaluation, and decision-making, according to the design objectives. It is generally undertaken during design review activities, where design outcomes are evaluated and decisions made to keep the project moving forward. Evaluation is the collaborative moment when team members elaborate a judgment by comparing the achieved solution with the design requirements and formulate possible changes.

Collaboration takes place along the lifecycle but assumes different characteristics. Germani et al. [38] recognize three main collaborative dimensions, differing in communication needs, tasks and interaction styles:

a. **Conceptual design collaboration** that anticipates the industrial design phase and aims at achieving the final product concept. It involves stylists and designers as industrial partners and project leaders and product managers as company's resources;
b. **Advanced design collaboration** that refers to product design phase and mainly consists of co-modelling activities. It engages engineers from technical department and quality department as well as the design and supply chain for assessing technical feasibility and economic convenience of every developed solution, engineering the validated product shape, developing manufacturing equipment and realizing all technical schedule for production.
c. **Interplay collaboration** that refers to data sharing and distribution and takes place in different moments along the product development process.

Such dimensions address the needs of different lifecycle phases; in a typical industrial design process they can be represented as in Fig. 7.2. It outlines a general product lifecycle focusing on the design stage and the main activities (in squared

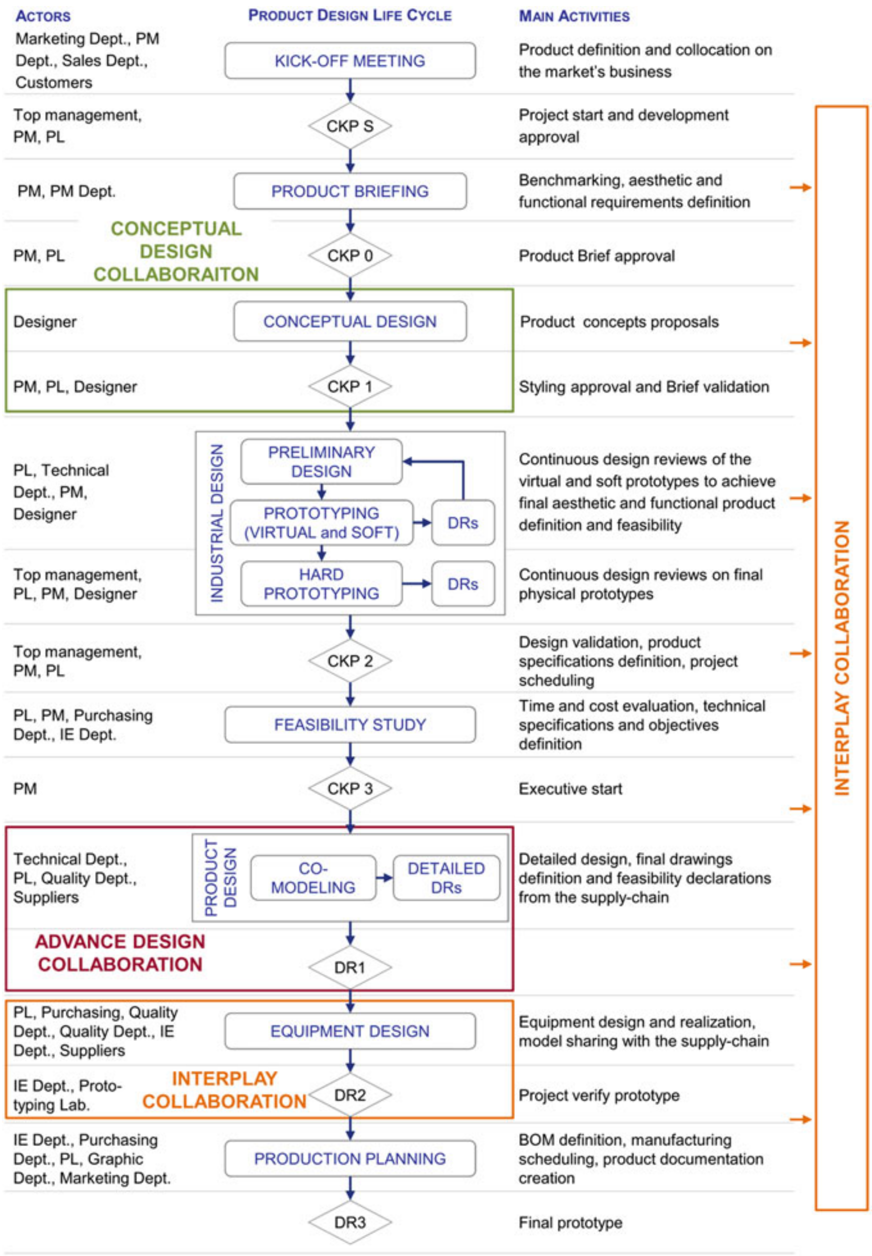


Fig. 7.2 Collaborative dimensions identified in a typical industrial design process [38]

blocks) and decisional points (in diamond-shaped blocks), and, for each of them, the main collaborative activities and the actors involved. It also highlights the different collaborative dimensions and elicits how they take place during the design lifecycle. In particular, crucial moments for collaboration are the so-called design review meetings (DRs), which occur during the entire process as highly collaborative moments when multidisciplinary team discuss, evaluate and decide to support the product design [31]. The number and occurrence, undertaken activities, representation modalities, supporting tools and team member skills vary according to the design stage to which they belong.

Thus, different DR forms can be recognized according to the process stage, design goals and outcomes, competencies involved, and exploited product representations: conceptual design collaboration and engineering design collaboration. During conceptual design collaboration the main conceptual aspects are simultaneously evaluated (e.g., aesthetics, function, ergonomics, cost, and product feasibility) and the competencies involved vary from decision-making staff to marketing operators, stylists, and technical managers. Usually, different product representations are used, both intuitive and technical (e.g. sketches, renders, rough physical prototypes, 2D drawings, functional schemes, 3D models, and digital mock-ups). Conceptual collaboration usually uses shared visualizations characterized by on-demand model sections, additional model views and mark-ups.

Another relevant common aspect is represented by the spatial-temporal characterization of collaboration. Indeed, different styles can be defined according to different time and space combinations: different timing and location can characterize groupware activities. Timing depends on whether participants act at the same or different times, i.e. synchronous or asynchronous collaboration. Location depends on where the participants are geographically, whether in the same place (co-located collaboration) or at different sites (remote collaboration) [39]. In product design four collaboration styles can be identified and separately analysed [40]:

- Synchronous and co-located (e.g. face-to-face meetings);
- Synchronous and remote (e.g. remote meetings between different sites);
- Asynchronous and co-located (e.g. routine design activity of a team inside the same company);
- Asynchronous and remote (e.g. routine design activity of a team involving multiple companies at different geographical locations).

Those styles are general and can be applied to the different dimensions of design collaboration. For instance, during conceptual design collaboration numerous human-to-human interaction events take place, as collaboration tends to be synchronous (co-located or remote).

The spatial dimension of collaboration introduces another crucial aspect: the distributed virtual collaboration within a "*temporary network of independent companies which cooperate in order to exploit the fast evolving business opportunity with the assistance of information technology to share expenses, skills and access to global market*" [41]. Indeed, when people are located in different sites and create the “virtual teamwork”, further issues emerge. Byrne et al. point out three crucial aspects:

- the importance of knowledge sharing integrating different independent organizational structures;
- the role of distributed information management to support inter-actions among conflicting competences and goals;
- the concept of temporary structures that should be reconfigurable according to different project tasks, product development stages and team members.

7.4.3 Product Models

In a collaborative design process, product models need to be disseminated in a broad scope. As they are the most important assets of product development companies, so companies are usually reluctant to share these models directly to avoid the leakage of the commercial secrets to competitors. This concern makes it difficult to realize the full potential and benefits of collaboration (see Chap. 18).

Cross-functional teams, which are the essence of collaborative networks, demand an intensive knowledge exchange process. Thus, Collaborative Engineering requires the support of computational frameworks that handle product models, from pictorial, symbolic and linguistic in the early stages, to virtual and algorithmic in the last stages [27]. Computational support for the generation and use of shared representations in Collaborative Engineering needs to include functional abstraction, geometric representation, constraints, and generation of multiple functional views through the product lifecycle. CAD systems, meanwhile, are moving beyond the representation of purely geometric entities, to integrating knowledge from the design and manufacturing domains into the CAD models as well [42] (see Chap. 10).

On the other hand, a product model is proprietary to a CAD system. To address these concerns, research efforts have been made to develop new representation schemes of product models based on open formats and neutral features (e.g. STEP, JT, eXtensible 3D, Web 3D, Universal 3D, Java 3D and OpenHSF). These representation schemes retain the essential visualization information of proprietary product models to support display-based manipulations, such as rotation and zooming, annotation, and mark-up. Major applications of these schemes for collaboration include customer surveys of product concepts and initial models, high-level project reviews among management, development and service departments, sales promotion, e-documents (e.g. Acrobat 3D), sharing catalogues, and visualization functions in Product Data Management (PDM) systems. Since only the visualization information is included in these schemes, crucial design information is protected (see Chap. 18).

There is an increasing interest for the use of 3D content due to the continuous development of visualization technologies. A real-time collaborative 3D virtual environment for multidisciplinary design is another growing research area where the scope of a problem is determined by exploring a range of alternative solutions to a brief or set of requirements in the conceptual design phase. Relevant challenges in this area are different decomposition schema of the model among the collaborators; relationships within and across the different schema; multiple representations and versioning of elements; ownership and access to elements and properties of elements and shared visual representation in a 3D virtual world (see Chap. 16). Chu et al. have described a scheme for collaborative 3D design using product models at various levels of details (LODs) [43]. Design features are selectively hidden at each level from certain participants, depending on their actual needs and individual accessibility in the collaboration. Therefore, the individual design know-how of the participating companies can be protected (see Chap. 18) [44].

7.4.4 Information Management

The need to incorporate more product lifecycle concerns imposes a stronger knowledge coupling of product development teams with diverse expertise [11]. Knowledge is generated by interaction between designers, teams, or organizations during the course of a product design, as well as the knowledge concerned with the objectives, processes, and results of disintegrating a product into two or more complex systems during the design processes, and re-integrating these systems to form the product. This requires not only the strategic predisposition to collaborate, but also the means to exchange of product data across different enterprises. So information sharing and semantic interoperability are vital keys for successful inter-enterprise collaborative design chains [44, 45].

The scenario is more complex when small and medium sized enterprises (SMEs) are part of the product development team, which is becoming increasingly frequent in industry. Indeed, in the development chain a lot of SMEs usually collaborate with the leader company to act on the same stages of design and production, from technical feasibility to assembly, from styling definition to maintenance [46]. Their involvement is particularly valuable as it allows big companies to lean their inner processes and to reduce personnel and costs. This fact contributes to a higher level of complexity when SMEs are involved in the design chain, given that: (i) they usually team up with other companies to support product design and manufacturing; (ii) they support more than one Large Manufacturing Enterprise (LME) at the same time; (iii) they are involved in different design stages; (iv) they should adopt specific and customized solutions; and (v) they usually use different software tools and representation modalities in order to perform specific tasks. Furthermore, the formalized in-house knowledge in SMEs is limited as well as human and financial resources respectively dedicated [38].

One of the most significant opportunities presented by collaboration is that of reconfiguration, which is the ability to reconfigure the overall capability, size and expertise of a business through strategic alliances with other complementary partners [10]. Reconfiguration can provide agility to help meet changing demands or respond to external environment events. SMEs can specifically become agile by forming dynamic partnerships to develop products that none of the individual partners could produce alone [47, 48].

Another important issue concerns with the management of the distributed knowledge across the design cycle, among teamwork members, inside different organizations involved in new product development. Indeed, proper distributed information management requires the creation of inter-enterprises business collaboration while keeping autonomy of participating enterprises [49].

Some attempts have been focused on the adoption of a workflow view approach to drive cross-organizational workflow interoperability. The aim is to efficiently manage dynamic and distributed design processes, comprehending all phases from marketing analysis, to conceptual design, to engineering, until validation and product release [50, 51]. These systems focus more on managing internal and external processes instead of investigating how the distributed knowledge can be formalized through the flow of activities or which are the best mechanisms for collective problem solving. They do not provide a shared environment where participants act to carry out different tasks according to the well-defined workflow model steps. In addition to problems of limiting companies to organize themselves to predefined structures, all available Workflow Management Systems (WfMSs) are unable to provide adequate answers to exceptions or deviations that can occur during the design process and that differ from the implemented process. In the last years, several researches have been oriented to the management of dynamic workflows by developing advanced frameworks to support exception handling and facilitate regaining model control after the exception has been resolved [52–54].

There are systems, on the other hand, that adopt a PLM-oriented perspective. They provide large-scale knowledge management tools, are strongly document-oriented, have a structured and not much customizable data model and suffer from inter-enterprise integration difficulties, inappropriate contextual reuse and fragmentation of knowledge [42, 55, 56]. They imply much effort to represent specific company's requirements and processes, to structure adequate knowledge-based databases and to integrate all existing digital technologies. Multi-agent systems have been introduced to support content management in product-process-design projects [57]. In recent years, attempts have been made to adopt web-based and cloud-based tools for tasks coordination, controlled knowledge sharing and information management [40, 58]. Nonetheless, confidentiality and security issues are inevitably present when it comes to cross-enterprise sensitive information sharing.

7.5 Selection of Collaboration Tools

The previous paragraphs described the main characteristics of design collaboration and the main technological solutions for collaborative work. However, despite the long research in this field, design collaboration is still poorly supported in industrial context. Indeed, such a process is complex and unstructured, and the collaboration needs variety according to the specific context of use and people involved.

In this context, selecting the most appropriate tools for enhancing collaboration in the specific design team and evaluating the achieved performance as well as their impact on industrial processes and human collaboration are fundamental [59]. The main challenge is the identification of the specific collaboration requirements and the definition of those characteristics that should be encompassed successfully by a specific CSCD tool when it is used in a collaborative venture that usually involves both SMEs and LMEs. The crucial point is the correlation between collaborative activities and tool functionalities by mapping functional and technical design features. A valid technique for correlation is Quality Function Deployment (QFD) [60], which has been proved to be successful for PLM systems benchmarking [61] and virtual reality technologies benchmarking [40].

In another work, Germani et al. [62] defines a structured method to specifically correlate collaborative design (co-design) activities with existing CSCD tools in order to select the most suitable technology for the specific context of application. In particular, it adopts a set of houses of quality (HoQ) to progressively associate collaborative activities with tool functionalities (at functional level) and with available software tool for their implementation (at technical level). Each HoQ is related to a specific aspect and its fulfilment allows populating the next HoQ; the process starts from collaboration needs, moves ahead with the mapping of co-design needs and tool functionalities, and ends with the weighting of the tools according to the specific needs.

The method can be outlined in five steps. Each one of them contributes to fulfil a HoQ and links to the next one:

1. Analysis of the product design process and recognition of the main interactions: the different collaborative dimensions are identified and the most critical stages are recognized and deepened;
2. Definition of a set of collaboration metrics to analyse collaboration performance and teamwork efficiency. For each metric, an evaluation weight is set according to the interaction mechanisms characterizing the specific collaborative dimension;
3. Correlation between the collaboration metrics and the co-design functionalities to define the technical requirements. The metrics express the collaboration needs to be satisfied during co-design processes whereas the functionalities represent the capabilities of different co-design technologies;
4. Classification of co-design tools according to the tasks achievement and the co-design functionalities;
5. Identification of the optimal tools to support the specific co-design process with respect to the related collaborative dimension (Fig. 7.3).

Such a method can be applied for both tool selection and performance evaluation. In the first case, the direct application of the proposed method (from step 1 to step 5) allows selecting a set of tools to create a suitable co-design platform maximizing both technical and functional performances; finally step 5 allows

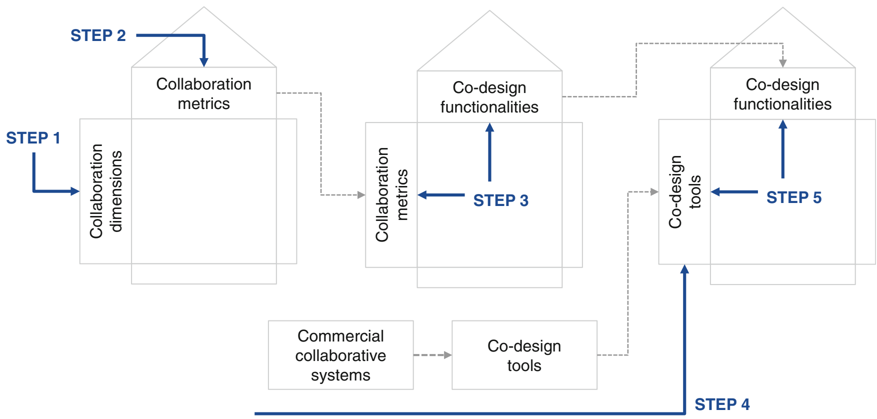


Fig. 7.3 Overview of the proposed method for co-design tools selection and evaluation [62]

benchmarking the technical performance of different commercial software solutions in accord with the identified functional requirements. In the second case, the method allows comparing the performances obtained with different tools to support the same co-design activity for a certain collaborative dimension; step 2 defines the evaluation metrics, step 5 maps the analysed tools functionalities and finally step 3 evaluates the tools' performance.

7.5.1 Metrics for Collaboration Performance and Teamwork Efficiency

In order to quantify the quality of collaboration that can be reached by adopting a certain CSCD tool during product design activities, a set of metrics is defined; they are grouped in three categories that point out three distinct aspects characterizing collaboration performances: task completion, team working and cognitive understanding. Table 7.1 provides a short description of the considered metrics and the categorization adopted.

Especially for benchmarking purposes, in order to have a reliable set of values for each dimension of collaboration, metrics are measured by collecting data from a large number of real processes and identifying five ranges of values that will be related to a five-point scale evaluation weights. Each cell of the correlation matrix (step 1, matrix 1) is represented by the average value expressed by (7.1):

$$\sigma_{ij} = \sum A_{ijk}/k \tag{7.1}$$

Table 7.1 Evaluation metrics to assess collaboration design performance in teamwork [40]

Topics	Collaborative tasks	Descriptions	Metrics
Task performance	Task control	Control of the project activities	Project errors or delays (nr)
	Workload	Organization ability and working time optimization	Fraction of time team is idle (%)
	Flexibility	Team ability to correctly plan resources and adjust activities when needed	Fraction of time schedule is adjusted—Supplementary DRs (%—nr)
	Synchronization	Team ability to work in synchronous and efficient way	Imperfect synchronization events (nr)
Team performance	Information acquisition	Team effort at collecting and acquiring project information	Fraction of time team is asked for information (%)
	Information provision	Team effort at providing necessary information	Fraction of time private information is provided to the group (%)
	Transfer of meaning	Team effort at efficiently communicating information to the group	Fraction of time information is provided (%)
	Brainstorming	Team ability to evaluate numerous alternative solutions	Decision alternatives considered (1/nr)
	Negotiating	Team ability to stimulate discussions and comparisons	Conflicting actions inside the team (nr)
	Critiquing	Team ability to stimulate comments and positive critiques	Comments and critiques on products (1/nr)
	Discovering differences	Team ability to identify differences and elaborate alternative solutions	Differences in understanding and elaborating identified (1/nr)
	Distributing	Team composition adequacy	People receiving information who should receive it (%)
Cognitive performance	Workload identifying	Accuracy of estimating workloads	Incorrect answers (%)

with $i \in [1, 17^{1}]$, $j \in [1, 3]$, and k = number of sample analysed processes. In order to relate nth metric value to the 5-point scale, it is necessary to calculate the average value m_{ij} and compare nth value with the five intervals defined by m_{ij}. It implies that n_{ij} value is proportional to A_{ij} score and high n_{ij} and σ_{ij} values have great importance. The whole process is summarized in step 1 and detailed in Fig. 7.4. When the

[1] Number of collaboration metrics.

					COLLABORATIVE DIMENSIONS		
					Conceptional design (1)	Advanced design (2)	Interplay (3)
COLLABORATION METRICS	TASK CONTROL	1	m1		σ11	σ12	σ13
		2	m2		σ21	σ22	σ23
		3	m3		σ31	σ32	σ33
		4	m4		–	σ42	σ43
	TEAM PERFORMANCE	5	m5	$n_{ij} \in [0 ; 2/m_i] \Rightarrow A_{ijk}=1$	–	–	–
		6	m6	$n_{ij} \in [2/5m_i ; 4/5m_i] \Rightarrow A_{ijk}=2$	–	–	–
		7	m7	$\forall\, n_{ij}$, if $n_{ij} \in [4/5m_i ; 6/5m_i] \Rightarrow A_{ijk}=3$	–	–	–
		8	m8	$n_{ij} \in [6/5m_i ; 8/5m_i] \Rightarrow A_{ijk}=4$	–	–	–
		9	m9	$n_{ij} \in [8/5m_i ; 2m_i] \Rightarrow A_{ijk}=5$	–	–	–
		10	m10		–	–	–
		11	m11	**Correlation between**	–	–	–
		12	m12	**ranges and 5-point scale weights**	–	–	–
	COGNITIVE PERFORMANCE	13	m13		–	–	–
		14	m14		–	–	–
		15	m15		–	–	–
		16	m16		–	–	–
		17	m17		–	–	

Fig. 7.4 Weighting of collaboration metrics for the three analysed collaborative dimensions [40]

sample enterprises are heterogeneous enough in tools exploitation, metric weights assume a general validity.

The following sections provide industrial case studies demonstrating how collaboration tools can be selected and applied in different design contexts.

7.6 Case Studies

The main challenge in collaborative design and engineering is still the performance evaluation for the specific context of use as well as requirements' satisfaction by the adopted supporting technologies. The crucial point is how to correlate collaborative activities with the features of the supporting tools and map functional and technical features. The research challenge is to identify which technology better answers to the specific needs of collaboration according to the lifecycle and actors involved (both SMEs and LMEs in virtual enterprises or collaborative networks).

7.6.1 The CO-ENV Co-design Platform to Support Industrial Chains

The CO-ENV case study refers to the definition of an innovative collaborative design platform for distributed industrial chains. In particular, it applied the above-mentioned methodology to benchmark the most suitable CSCD technologies to satisfy the needs of the CO-ENV project [63], which involved three of the main sectors of Italian industry. It engaged 21 Italian companies (5 LMEs and 16 SMEs), associated in a consortium and organized into three vertical industrial chains according to the product design goals: woodworking machines, household appliances and wellness products. Chain structures are quite similar and well represent a common linear model centred on one leader company [48]. This leader company conceives, realizes and commercializes the final product, while its partners provide professional services and supply components by intervening at different moments of the product development [64].

The case study focused on defining a web-based platform to support inter- and intra-chain collaboration for the three chains. The study lasted 1 year and involved two experts in product design process and two experts in CSCD systems coming from both industry and Academia to elicit the exchanged knowledge, to define the metrics' weights and to fill out HoQ matrices. Almost 30 projects have been analysed during the case study: they differ from product typology (bathtubs, shower trays, kitchen ovens, fridges, washing machines, dishwashing machines, etc.), activity goals (concept design of a new product, restyling of an existing product, design of a new family line, ideation of a new product for an existing line, usability optimization, manufacturing follow-up, etc.), collaborative dimensions (conceptual design, advanced design, interplay), actors involved (designers, stylists, technicians, marketing staff, technical engineers, industrial engineers, quality dept. staff, consultants from research institution, scientific partners, assembler subcontractors, moulding subcontractors, etc.). Teamwork monitoring allows defining the evaluation weights for each collaborative dimension considered by the method (step 1) and assessing the metrics importance for each co-design functions (step 2). Because of the complexity of the investigated processes, also completion times and companies' requirements have been considered, so that metrics' weights and importance have been defined by correlating the obtained performances with the expected ones in terms of time and results. In order to do this, collaborative tasks have been evaluated in a set of sub-activities, belonging to the product development process, with respect to the specific collaborative dimensions, as described in Table 7.2.

The CO-ENV case study allowed collecting a huge quantity of data about the development of collaborative industrial processes. In particular, attention has been focused on advanced design that was prominent dimension for the chains analysed. As a consequence, a set of weights for the collaborative metrics was defined for the CO-ENV companies. After that, the tools selection was carried out according to step 3 of the proposed method. In particular, it correlates collaboration needs with technical co-design functionalities and co-design tools' characteristics. For the CO-ENV purposes, functionalities have been grouped into seven categories: *visualization*, *project management*, *workflow management*, *annotation and analysis*, *data sharing*, *human collaboration*, *data control*.

The study highlighted that one functionality class was not enough to satisfy the CO-ENV needs. For example, data sharing capabilities must be related to simultaneously viewing and sharing of general documents and CAD files; data control tools are useful to list and track mark-ups and comments, to create a data repository

Table 7.2 Collaboration sub-activities monitored for the CO-ENV case study [63]

Collaborative dimensions	Collaborative sub-activities	Collected data
Conceptual design collaboration	1. Marketing requirements discussion, during which the defined marketing needs are discussed in critical way or evaluated by hindsight	Completion time
	2. Brief discussion, where team members discuss about the main brief statements and anchors	
	3. Product concept definition, during which the conceptual basic idea of the product is defined by evaluating different alternatives and considering the marketing and brief requirements	
	4. Styling approval, during which a preliminary aesthetical and technical feasibility is outlined and the product brief validated	
Advanced design collaboration	1. Co-modelling, during which different actors involved in product design work on the same product model in almost parallel way	Completion time
	2. Detailed design definition, during which cycling design reviews take place in order to produce the final bi-dimensional drawings and the detailed design project	
	3. Prototypes evaluation, where virtual and physical prototypes are evaluated and validated from different points of view (styling, functional, technical, reliability, usability, costs, assembleability, maintenance, etc.)	
	4. Feasibility analysis, where a detailed analysis is performed and the relative feasibility declarations are collected by the supply-chain	
Interplay collaboration	1. Development approval, that requires an informative data exchange	Completion time
	2. Benchmarking, during which data about the market and the main competitors' products need to be exchanged	
	3. Co-modelling, that requires an intense data sharing	
	4. Prototypes validation, where prototypes need to be viewed and analysed by many different competencies belonging to the project team	
	5. Feasibility analysis, where the leader company need to interface the suppliers and exchange product data and quotes with them	
	6. Equipment design, during which the leader company need to share product information with its sub-contractors	
	7. Product documentation creation, where both technical and commercial product information need to be shared by technical dept., graphics dept., marketing dept. and external providers (graphic studies, printers, etc.)	

and to search data and documents; workflow management facilities are important in supporting activity notification and handling unexpected events; visualization facilities are welcome to open and view different types of documents; finally, human collaboration capabilities required audio-video conferencing.

After having defined the technical requirements, the tools selection was carried out by considering the different tools commercially available and mapping their functionalities with the required ones. For this purpose, the most common CSCD tools were grouped in five functional categories: collaborative CAD systems (e.g. Oracle Autovue™, PTC Co-Create One Space Live!™, etc.), collaborative portal servers (e.g. Microsoft Sharepoint™, Mindquerry™, Cyn.in™, Plone™, etc.), collaborative PLM suites (e.g. cPLM™ solutions by PTC, Siemens, IBM, etc.), digital note software (e.g. Microsoft OneNote™, Post-it Digital Notes™, etc.) and communication tools (e.g. Skype™, Elluminate™, Messenger™, etc.). For each category, experts and managers from CO-ENV companies expressed their own opinions for the evaluated tools.

The case study highlighted that a single application was not enough to efficiently support complex collaborative scenarios, especially for industrial applications. It means that different types of co-design tools need to be adopted and single applications require to be integrated in a common environment.

In this context, the proposed method allows identifying the main co-design modules that should be considered and properly integrated. On the basis of the experimental results, a preliminary platform was defined.

The platform architecture adopted a client-server approach for reaching better performance during collaboration and having a common vault as shared data repository. The architecture consisted of three main components dedicated to different functions:

- Oracle Autovue™ as collaborative CAD tool for responding to visualization, annotation and data sharing priorities;
- Microsoft Sharepoint™ 2007 as collaborative portal server for managing projects and workflows data and information control;
- Skype™ as communication tool for supporting human collaboration.

By adopting such a collaborative platform, users could instantly access any supported document type from any desktop via the web regardless of their desktop configuration and without any client installation. Furthermore, they could simultaneously review and mark-up documents, even technical ones such as CAD files, assign action items and resolve design issues in real-time. In particular, three areas were created:

- Project Area: the operative collaboration interface for any actor participating to almost one of the CO-ENV projects. It is supported by the collaborative portal server and a common vault. The portal server can easily manage server applications, so users can contemporarily and independently access to the server and enter different areas properly organized. All projects have their own area where similar and customized functionalities are implemented;

- Co-design Area: allows collaborating on shared product models and calls back external applications (Oracle Autovue™ and Skype™). It can be entered directly on server through the collaboration area in the corresponding project space. Users are able to exploit the specific software applications and recover data and documents from the common vault at the same time. The vault also stores all co-design session reports and data indexing. As a result a bi-directional information flow is created between the two environments;
- Administrative Area: the operative administration interface used by the CO-ENV companies. It is supported by the portal server and it recovers data from the common vault as well as the project area.

7.6.2 The DesigNET Collaborative Platform for Contract Design

This case study adopted a similar methodology of the previous case study, but the final purpose was completely different since it aims to create a collaborative platform to support design activities for contract design, which is characterized by different actors, organization, scopes and constraints.

The methodology has been applied to support a cluster of companies operating in the contract furniture sector within the DesigNET project [65]. DesigNET focuses on hospitality and retail contract design and aims to promote Made in Italy innovation and lifestyle by creating a multi-disciplinary organization and offering co-designed integrated solutions. The project started in 2011, involved 17 Italian companies (from product manufactures, to suppliers, architects and design studios) that vary in size, organization and core business, and was funded the Italian Ministry of Economic Development. The goal was to realize a collaborative network thanks to an innovative technological platform able:

- to showcase the DesigNET companies innovations and competencies,
- to configure the designed space as a whole and the single products in details to meet commitment expectations and companies' capabilities,
- to design custom products, personalized variants or new integrated solutions in an effective collaborative way.

Also in this case, the tool selection was carried out according to the correlation matrices. However, since contract design is a novel and almost unexplored field in design, the study also analysed the AS-IS process to identify the requirements of collaboration. So the method was derived from the general one but was enhanced with process investigation and consisted of the following steps:

1. Analysis of the AS-IS contract design process by questionnaires and direct interviews: it allows modelling the actual process by mind maps and highlighting the main criticalities. Analysis concerns the process activities development and

tasks, the collaboration issues, input and output data typology and management, design offer features and variability. All feedbacks are collected and the most frequent responses are considered;

2. Elicitation of collaborative requirements: a set of expected requirements are elicited from the previous analyses and each requirement is provided by a weight (5-point scale) expressing its relevance according to both experts and process actors feedback;
3. Benchmarking of the supporting technologies: carried out according to the general methodology and aims to identify the most suitable technologies. Benchmarking exploits a set of HoQs to correlate collaboration requirements and system capabilities.

The AS-IS process analysis outlined the main characteristics of the contract furniture process and the research challenges. Also in this case investigation is carried out by questionnaires and interviews and is guided by experts from Academia and industry. After that, the DesigNET companies have been involved both in requirements elicitation and weights assessment. For each company, two mangers from R&D department and from the marketing department are asked to express the importance that each requirement has for the company but also for the architects and designers he/she collaborates with in contract furniture. In addition, five external designers, that usually works in hospitality and retail, and two general contractors are involved in this assessment. Weights data are averaged on 40 total judges. A list of requirements has been defined.

A preliminary review of potential supporting software tools for furniture design has led to the following classification:

- CAD-based configuration systems: they refer to commercial systems or open-source platforms dedicated to the furniture sector, e.g. Metron™ (http://tesysoftware.net), 3CAD evolution™ (http://www.3cadevolution.it), Mobilia™ (http://mobiliasoft.com). They are client-based and allow handling CAD models and configuring them and generating the complete BOM. They adopt a single company perspective, so they cannot support co-design within an extended network;
- General-purpose 3D modelling systems: they are 3D modelling tools mainly adopted for architectural design, e.g. Google Sketch-up™ and Sketch-up PRO™ (http://www.sketchup.com/intl/en). They are general-purpose and easy-to-use, so they can be easily adopted to create an environment and populate it with product models. They are client-based, but some of them support model sharing through the web. However, rendering quality is low and most design tasks are not fully supported;
- Web-based 3D configuration systems: they are free or open-source platforms for interior design that allow creating a 2D-3D environment where furniture items can be positioned and rendered, e.g. Sweethome3D™ (http://www.sweethome3d.com), DomusPlanner™ (http://www.domusplanner.com). They are intuitive, low cost, and allow data sharing through the web. However, they

do not support technical product configuration so that they are not suitable for architects and/or manufacturers;

- Web-based visualization tools: they afford to visualize 3D models in a shared modality by multiple users, navigate the space also by walkthrough, mark-up file, chat and comment during designing, e.g. Oracle Autovue™ (http://www.oracle.com/us/products/applications/autoVue), Actify SpinFire™ (http://www.actify.com/products/spinfire-cad-solution-system), Autodesk Streamline™ (http://www.autodesk.it/streamline). They are low-cost and multi-systems, but image quality is low and real time modelling is not usually supported;
- CAD-based plug-in for configuration management: they are plug-in applications developed for specific CAD commercial systems (i.e. SolidEdge™, SolidWorks™, PRO/ENGINEER™, CATIA™) to manage product variables and assembly configuration, create relationships among product features and dimensions, and handle modular assemblies. They fully support design tasks, but rendering quality is poor and data sharing is not available. They are not easy to use for non-expert users.

Beyond these tools, there are several IT development frameworks and platforms that allow realizing specific applications for high-quality rendering or data management purposes, e.g. .NET (http://www.microsoft.com/net), X3D (http://www.web3d.org/x3d), OpenGL (http://www.opengl.org), and JReality (http://www3.math.tu-berlin.de/jreality).

The benchmarking involved two experts, one from Academia and one from the largest company of the partnership evaluate each tool (j) assigning 0–3–9 values (Bi) for each user requirements (i). For each tool values are weighted according to the requirements' relevance (Ai) and then summed according to Eq. (7.1) to obtain a total evaluation.

The selected technologies were properly integrated to realize a unique system accessible by different user interfaces supporting diverse viewpoints and levels of abstraction. Indeed, as the stakeholders differ for their personal background (e.g. engineering, architectural, economics), purposes (e.g. technical, economical, product-centred, holistic, etc.) and needs (e.g. the interior architect has to configure the space, the designer to shape a new customized product, the contractor/buyer to find out the cheapest solution and have a global overview of the furniture offer and the manufacturer to create an offer based on user requests) it is imperative to provide at least four user interfaces with different functionalities and levels of usability. The platform architecture is structured in two main modules whose access is provided by the different user interfaces. Main input and output data are defined and then organized into a unique system platform, which has four main interfaces:

a. *Virtual Catalogue*: it is a web-based marketplace where the user can view a rich catalogue of products and integrated solutions proposed by manufacturing companies and evaluate all product variables by a high-quality 3D rendering, refreshing once a parameter changes (e.g. colour, finishing, dimensions, accessories, performance). Each item is correlated with its technical documentation (e.g. 3D models, 2D drawings, manuals, data sheet). Such interface is barrier-free since it is on the web and has a public access throughout the most common Internet browsers;

b. *Interior Design Configurator*: it is the configuration engine and allows the user to create a personal project, importing a 2D or 3D space model (e.g. hotel room, store space) and populate the empty space by selecting the catalogue items. It supports product configuration and positioning into the space by following the manufacturer's guidelines (e.g. a bookcase that requires to be attached to the wall) and respecting some technical constraints (e.g. the minimum empty surface of the wall and its minimum resistance) by exploiting a knowledge-based set of rules which explicit the relationships among products and the environment;
c. *Data Manager*: it is a technical product configurator that allows the manufacturer to upload and define their own products, all feasible variables and the possible ranges of parameters' modification (see Chap. 14). For each item the company has to provide a 3D model, indicate the product existing or customizable characteristics (e.g. materials, surface finishing), specify the optional accessories (e.g. handles typology), define the customizable features and their allowed range of variation (e.g. max-min length), add the installation constraints to be respected (e.g. maximum distance to the power socket of 30 cm) or the suggested configuration constraints (e.g. wall contact is required). Additional data can be further attached such as user manuals or product renderings;
d. *Co-Designer*: it supports the technical configuration and co-creation of customized products or integrated design solutions by a web-based collaborative space. Such a tool is fundamental when the existing products cannot satisfy a certain demand and a customized product is required or when the architect has to create a special solution for the configured space.

7.7 Conclusions and Outlook

Effective collaboration can help resolve conflicts early in the design stage and reduce product development lead-time and manufacturing costs [5, 66]. However, what will organizations benefit, if embarking in a collaborative network? Will the benefits compensate for the extra overhead, losing some control, and even taking the risks that collaboration implies? These are main questions that many small and medium enterprise (SME) managers ask when the issue of collaboration is brought up [9, 46].

Collaboration involves considerable preparation costs/time, in addition to the operational overheads and risks, which represent barriers to the rapid formation of dynamic coalitions in response to business opportunities [48]. As a basic rule, in order to support rapid formation of collaborative networks, it is necessary that potential partners are ready in advance and prepared to participate in such collaboration. This preparedness includes compliance with a common interoperable infrastructure, adoption of common operating rules, and common collaboration agreement, among others [67]. Any collaboration also requires a base level of trust among the organizations [68].

The industrial cases overviewed in this chapter indicate that by planning a collaborative platform (i.e., with complementary software applications), users can best benefit from the most recent technology developments for establishing meaningful and timely collaboration with minimum technological hindrances and steep learning curves.

With the globalization of the manufacturing industry and the development of worldwide production consortia, cultural problems have attracted much attention [6]. Collaborative design systems still need to integrate results from human sciences in order to address the cultural differences, not only between designers and product users, but also among other stakeholders [68, 69]. The socio-technical environment of product design and realization is yet to be fully understood and modelled, to give way to computer-based systems that can mimic and facilitate this, which is one of the most valuable human behaviours: collaboration (see Chap. 6).

References

1. Argandoña A (1998) The stakeholder theory and the common good. J Bus Ethics 17:1093–1102
2. Warry JG (2000) Warfare in the classical world. Barnes and Noble, New York
3. Lozano R (2008) Developing collaborative and sustainable organisations. J Clean Prod 16:499–509
4. Todd S (1992) Collective action: theory and application. University of Michigan Press, Ann Arbor
5. Lu SCY, Elmaraghy W, Schuh G, Wilhelm R (2007) A scientific foundation of collaborative engineering. CIRP Ann Manuf Technol 56:605–634
6. Shen W, Hao Q, Li W (2008) Computer supported collaborative design: retrospective and perspective. Comput Ind 59:855–862
7. Gaver WW (1991) Sound support for collaboration. In: Proceedings of the second conference on European conference on computer-supported cooperative work (ed). Kluwer Academic Publishers, Amsterdam, pp 293–308
8. Mattessich PW, Murray-Close M, Monsey BR (2001) Collaboration—what makes it work. Amherst H. Wilder Foundation, St. Paul
9. Grandori A (1997) An organizational assessment of interfirm coordination modes. Organ Stud 18:897–925
10. Camarinha-Matos LM, Afsarmanesh H, Galeano N, Molina A (2009) Collaborative networked organizations—concepts and practice in manufacturing enterprises. Comput Ind Eng 57:46–60
11. Lomas CDW (2009) A design framework for agile virtual enterprise collaboration. PhD thesis, Durham University
12. Lu SCY, Li Q, Case M, Grobler F (2006) A socio-technical framework for collaborative product development. J Comput Inf Sci Eng 6:160–169
13. Morgan JM, Liker JK (2006) The Toyota product development system. Integrating people, process, and technology. B&T, Charlotte
14. Migliarese PaCV (2006) Cooperation and coordination in virtual enterprises: the role of e-collaboration tools. In: Information systems and collaboration: state of the art and

perspectives—best papers of the 11th international conference of the association information and management, AIM 2006, pp 173–86
15. Galbraith JR (1995) Designing organizations. Jossey Bass Publishers, San Francisco
16. Martinez MT, Fouletier P, Park KH, Favrel J (2001) Virtual enterprise—organisation, evolution and control. Int J Prod Econ 74:225–238
17. Thompson K (2005) A taxonomy of virtual business networks. The bumble bee. http://www.bioteams.com/2005/07/18/a_taxonomy_of.html. Accessed 15 Feb 2014
18. Le Q, Panchal JH (2011) Modeling the effect of product architecture on mass-collaborative processes. J comput Inf Sci Eng 11(1):011003
19. Weiseth PE, Munkvold BE, Tvedte B, Larsen S (2006) The wheel of collaboration tools: a typology for analysis within a holistic framework. In: Proceedings of the ACM conference on computer supported cooperative work, CSCW, pp 239–48
20. Greif I (1988) Computer-supported cooperative work: a book of readings. Morgan Kaufmann, San Mateo
21. Martínez-Carreras MA, Muñoz A, Botía J (2013) Building and evaluating context-aware collaborative working environments. Inf Sci 235:224–241
22. Eynard B, Lienard S, Charles S, Odinot A (2005) Web-based collaborative engineering support system: applications in mechanical design and structural analysis. Concur Eng Res Appl 13:145–153
23. Reddy DR, Erman LD, Fennell RD, Neely RB (1976) The Hearsay-I speech understanding system: an example of the recognition process. Comput IEEE Trans 100:422–431
24. Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. Comput IEEE Trans 100:1104–1113
25. Norrie DH, Kwok A (1991) Object-oriented distributed artificial intelligence. In: New results and new trends in computer science. Springer, Heidelberg, pp 225–242
26. Shen W, Hao Q, Yoon HJ, Norrie DH (2006) Applications of agent-based systems in intelligent manufacturing: an updated review. Adv Eng Inf 20:415–431
27. Chandrasegaran SK, Ramani K, Sriram RD, Horvath I, Bernard A, Harik RF et al (2013) The evolution, challenges, and future of knowledge representation in product design systems. Comput Aided Des 45(2):204–228
28. Li W, Ong S, Nee A (2006) Integrated and collaborative product development environment: technologies and implementations. World Scientific Publishing, Singapore
29. Zhang M, Chen Y (2013) Collaborative design theory and related key technology study based on cloud computing. J Softw Eng Appl 6:18
30. Valilai OF, Houshmand M (2013) A collaborative and integrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm. Rob Comput Integr Manuf 29:110–127
31. Lloyd McDonnell J (2009) Introduction to about: designing-analysing design meetings. In: McDonnell J, Lloyd P (eds) About: designing—analysing design meetings. CRC Press, Boca Raton
32. Pol G, Merlo C, Legardeur J, Jared G (2007) Collaboration in product design and PLM-based coordination. In: 4th, international conference on product lifecycle management; product lifecycle management: assessing the industrial relevance. Inderscience Publishers, Geneva, pp 21–30
33. Noble D, Letsky M (2002) Cognitive-based metrics to evaluate collaboration effectiveness. In: Analysis of the military effectiveness of future C2 concepts and systems, NC3A, The Hague, The Netherlands, 23–25 Apr 2002, and published in RTO-MP-117. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA425450. Accessed 15 Feb 2014
34. Norman DA (2013) The design of everyday things. Basic Books, New York
35. Tripathy A, Eppinger SD (2011) Organizing global product development for complex engineered systems. IEEE Trans Eng Manage 58(3):510–529
36. CIMdata (2010) Product lifecycle management (PLM) definition. http://www.cimdata.com/plm/definition.html. Accessed 25 Feb 2014

37. Kiritsis D, Bufardi A, Xirouchakis P (2003) Research issues on product lifecycle management and information tracking using smart embedded systems. Adv Eng Inf 17:189–202
38. Germani M, Mengoni M, Peruzzini M (2011) How to address virtual teamwork in SMEs by an innovative co-design platform. Int J Prod Lifecycle Manage 5:54–72
39. Barratt M (2004) Understanding the meaning of collaboration in the supply chain. Supply Chain Manage Int J 9:30–42
40. Germani M, Mengoni M, Peruzzini M (2012) An approach to assessing virtual environments for synchronous and remote collaborative design. Adv Eng Inf 26(4):793–813
41. Byrne J, Brandt R, Port O (1993) The virtual corporation. Bus Week 8:98–103
42. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manage 7(3/4):324–347
43. Chu C-H, Wu P-H, Hsu Y-C (2009) Multi-agent collaborative 3D design with geometric model at different levels of detail. Rob Comput Integr Manuf 25:334–347
44. Wang J, Chu D (2012) Survey on collaborative awareness model for CSCW and trends of the new age. In: 2nd international conference on mechanical engineering and green manufacturing, MEGM 2012, 16–18 Mar 2012, vol 155–156. Trans Tech Publications, Chongqing, pp 357–362
45. Teixeira KC, Borsato M (2014) Semantic modeling of dynamic extended companies. In: Cha J et al. (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc international conference on concurrent engineering. IOS Press, Amsterdam, pp 215–224
46. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manage 7(2):116–131
47. Yusuf YY, Sarhadi M, Gunasekaran A (1999) Agile manufacturing: The drivers, concepts and attributes. Int J Prod Econ 62:33–43
48. Moynihan P, Dai W (2011) Agile supply chain management: a services system approach. Int J Agile Syst Manage 4(4):280–300
49. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manage 7(2):101–115
50. Schulz KA, Orlowska ME (2004) Facilitating cross-organisational workflows with a workflow view approach. Data Knowl Eng 51:109–147
51. Jiang P, Shao X, Qiu H, Gao L, Li P (2009) A web services and process-view combined approach for process management of collaborative product development. Comput Ind 60:416–427
52. Luo Z, Sheth A, Kochut K, Miller J (2000) Exception handling in workflow systems. Appl Intell 13:125–147
53. Mourão H, Antunes P (2007) Supporting effective unexpected exceptions handling in workflow management systems. In: Proceedings of the 22nd annual ACM symposium on applied computing, pp 1242–1249, ACM Press. http://www.di.fc.ul.pt/~paa/papers/sac-07.pdf. Accessed 15 Feb 2014
54. Ito T (2014) A proposal of body movement-based interaction towards remote collaboration for concurrent engineering. Int J Agile Syst Manage 7(3/4):365–382
55. Aziz H, Gao J, Maropoulos P, Cheung WM (2005) Open standard, open source and peer-to-peer tools and methods for collaborative product development. Comput Ind 56:260–271
56. Therani M, Tanniru M (2005) Knowledge partitioning: a strategic approach to product lifecycle management. Int J Prod Dev 2:85–108
57. Gomes S, Monticolo D, Hilaire V, Eynard B (2009) Content management based on multi-agent system for collaborative design. Int J Prod Dev 8:178–192
58. Pollalis YA, Dimitriou NK (2008) Knowledge management in virtual enterprises: a systemic multi-methodology towards the strategic use of information. Int J Inf Manage 28:305–321
59. Carvalho H (2013) An innovative agile and resilient index for the automotive supply chain. Int J Agile Syst Manage 6(3):258–278

60. Cohen L, Cohen L (1995) Quality function deployment: how to make QFD work for you, Addison-Wesley, Reading
61. Benassi M, Bordegoni M, Cascini G (2004) A methodology for evaluating the efficiency of PLM tools in product development processes. In: Hinduja S (ed) Proceedings of the 34th international MATADOR conference. Springer, London, pp 373–378
62. Germani M, Mengoni M, Peruzzini M (2012) A QFD-based method to support SMEs in benchmarking co-design tools. Comput Ind 63(1):12–29
63. Collaborative environments for innovation on the territory (COENV) project. www.coenv.it. Accessed 15 Feb 2014
64. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manage 6(4):307–323
65. Contract design (DESIGNET) project. www.designet-italy.it. Accessed 15 Feb 2014
66. Peruzzini M, Germani M (2014) Design for sustainability of product-service systems. Int J Agile Syst Manage 7(3/4):206–219
67. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Syst Manage 6(1):7–24
68. Chang D, Chen CH (2014) Understanding the Influence of Customers on Product Innovation. Int J Agile Syst Manage 7(3/4):348–364
69. Chang D, Chen CH (2014) Exploration of a concept screening method in a crowdsourcing environment. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc international conference on concurrent engineering. IOS Press, Amsterdam, pp 861–870

Chapter 8
Design of Complex Programs as Sociotechnical Systems

Bryan R. Moser and Ralph T. Wood

Abstract Following the introduction of systems thinking concepts in Chap. 3, we demonstrate here the treatment of complex engineering projects as sociotechnical systems in practical engineering practice. This approach, called Project Design, enables concurrent engineering (CE) teams to foresee the influence of project architecture, behaviors, dependencies, and complexity on emergent performance, thereby reducing the occurrence of unpleasant surprises. We have seen in multiple industrial cases this method as a source of new thinking and practices relevant to CE, with supporting tools and processes. Past assumptions about standard work practices may be tested, including such factors as degree of concurrency, phasing, roles, technology decomposition, system interfaces, and risk and its reduction. If embedded behaviors, in interplay with the total project architecture, lead to surprising negative or positive performance, the design of the engineering project as a sociotechnical system begins with un-learning, then awareness, and then learning of the project approaches more likely to produce positive results. The design of concurrency is specific to the nature of the social and technical elements of the system and its architecture.

Keywords Sociotechnical systems · Project design · Collaborative engineering · Simulation-based planning · Scheduling · Complexity · Teamwork · Learning

B.R. Moser (✉)
Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: bry@mit.edu

B.R. Moser · R.T. Wood
Global Project Design, 75 Arlington Street Suite 500, Boston, MA, USA
e-mail: ralph.wood@gpdesign.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_8

8.1 Introduction

Leveraging systems thinking for the planning and performance of concurrent engineering (CE), we build project models, simulate to forecast likely outcomes, and lead cross-functional teams to explore and converge across multiple scenarios in planning work-shops. Rather than finding a single, optimal process which assumes stability and absence of real-world uncertainty of cost, schedule, and quality, we capture the characteristics of the project that will allow insights about feasibility, value, and a tradespace of likely outcomes.

We refer to this integrated approach as Project Design—leveraging sociotechnical systems thinking—relying on:

- **representation**: sociotechnical system models
- **analytics**: behavior based simulations
- **workshops**: a social process of engagement, trade-offs, and learning

Together these three enable CE teams to expose deeply embedded assumptions of expected performance, test multiples alternatives project architectures and behaviors, and more rapidly converge on feasible and optimal plans.

This chapter begins with an overview of these three components of the Project De-sign approach. Representation is described as modeling of sociotechnical characteristics include scope, teams, complexity, distribution, coordination as activity, and con-current and mutual dependence. Next, agent-based simulation of the engineering project is described which analyzes realistic outcomes including limits due to technical complexity, human priorities, capacities, interactions, and mistakes. These models and analytics are then shown as they are used in collaborative workshops. Two industrial cases are discussed, followed by a comparison with contemporary approaches and discussion of the benefits of Project Design.

8.2 Representation: Models of Engineering Projects

Our efforts to better represent the sociotechnical characteristics of product development projects began in 1995 at the University of Tokyo [1]. A model represents how a project is structured, products characterize the scope, teams are assigned and prioritize work and coordination, and progress is realized through activities. Progress is constrained not only by the capacity of teams but also by dependencies amongst activities, phases, and products. The total form of the project—including model elements and relationships—constitutes the architecture of the project.

An effective model of an engineering project as a sociotechnical system captures the essence rather than details of product, process, and organization—and especially how they interrelate. Rather than detailed decomposition of tasks, we emphasize a higher level representation of architecture and the interactions across these model elements.

 **Product**	*Products* are meaningful results of completed work. The product system is most likely the linkage to overall project value. A product is realized through activities, which represent kinds of scope and progress.
 **Team**	*Teams* are one or more people who make effort to work and coordinate by applying abilities. Teams work on activities through contracts to indicate that the team has some role in that Activity.
 **Phase**	*Phases* are grouped activities that represent flow of progress over time. These phases are stages of scope and progress which may stretch across multiple products yet are viewed together for governance.

Fig. 8.1 Three fundamental element types cover product, process, and organization

These three project element types are grouped into breakdown structures—outcome centric product breakdown structure (PBS), teams grouped into a project's organization breakdown structure (OBS), and the phase based groups which often form the topmost level of a work breakdown structure (WBS). Product, phases, and teams are three points of view from which the sociotechnical models are seen and evolve (Fig. 8.1).

In contrast to common Gantt and PERT charts which emphasize task-based processes, a sociotechnical model retains the distinction in aggregate between product and process, that is, between results and the flow to complete these results.

Often in industrial use of traditional scheduling the detailed WBS describes a mix of these concepts; some parts of the WBS in a master schedule are product centric while other parts are phase or team (functional) centric. In those cases the mapping of scheduled activities to the product, governing process, or project team organization can be unclear.

A project activity is an effort by teams to generate part of a product during a phase; the activity ties these other elements together (see Fig. 8.2) Activities are described with scope as a unit of progress (e.g. drawings, parts, and tests), abilities

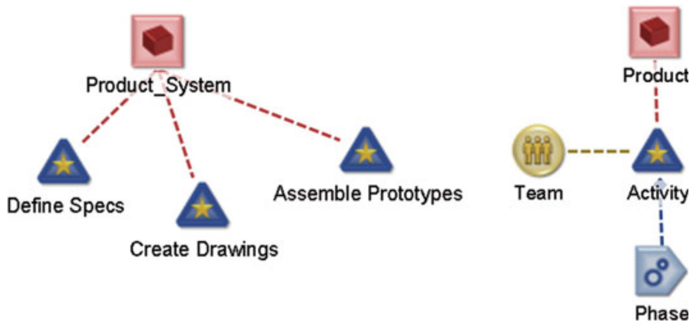


Fig. 8.2 Activity as product scope (*left*) and leaf of three breakdown structures (*right*)

required, nominal work effort, and complexity. Any work or coordination related to the activity is carried out by people on teams using resources. Multiple structures overlap and are linked by various relationships as a natural characteristic of the project, yet contrast with idealized process views.

The fact that multiple structures overlap and are linked by various relationships is a natural characteristic of the project, yet contrast with an idealized process task view. Referring to a PERT chart showing an ideal development process, Bucciarelli wrote:

> To anyone interested in process, these diagrams shed very little light on how design acts are actually carried out or who is responsible for each of the tasks within various boxes. Nor is it apparent what these participants need to know, what resources they must bring to their task, and, most important, how they must work with others. The lines with arrows hardly represent the negotiation and exchange that go on within designing [2].

To better shed light on real-world activity in engineering projects, the next sections discuss our method's representation of complexity, coordination, and dependencies which improve the integrity of these models.

8.2.1 Complexity in Project Activities

What makes a project activity complex? High complexity activities tend to require more information sharing and rework. Low complexity activities tend to proceed more smoothly, with limited need for coordination and rework. While some view complexity as a purely technical characteristic, when viewing engineering as a sociotechnical system the complexity is also driven by the condition of teams. Some refer to this human derived measure as "complicatedness", with the phrase "complex" an inherent property of the physical system. In our case, we refer to the complexity of the total sociotechnical system.

Engineering project complexity is defined here as *the cost of uncertainty reduction*—the gap between information as known by humans (tacit and transferable within a practical time horizon) and information of value to the project in the external world. A product system that is fully explored and known is no longer complex, if the knowledge is available at low cost. Complexity is therefore a function of the condition of information in technical and human forms—in current products, standards, engineering systems, and as well as the state of knowledge of the human teams. With over 15 years of industrial application, we continue to find ways to assess, sample, and inquire about the information condition of specific products, processes, and teams, allowing for calibration of complexity measures across an actual project. In turn, for effective Project Design, assessment of complexity is not necessary across all pockets of a program, rather in those which have more significance for systemic outcomes.

8.2.2 Coordination as Activity

The dynamics of contemporary engineering projects are driven by team activities beyond their own direct work, especially coordination. Brooks cites project group failures as stemming from the "man-month" conception, the notion that the number of individuals assigned to a project per month is an appropriate way of determining its dimensions. Consistent with Brooks' man-month argument, Holt [3] states that coordination is the "greatest common denominator" of group activities and asserts that, despite its importance, coordination is an "odd category" of activity because it has no direct product, it often cannot be performed alone, and much of it is not even performed consciously. Since coordination is "odd" in these respects, it receives less than its due share of management's attention (although it consumes a large share of organizational resources) and has not as yet obtained adequate research treatment [4].

Complex engineering projects are diverse and uncertain. Distributed teams working with little shared background across boundaries may over a long period develop their own work culture and thus approach efficiencies of a local project team. However, some increase in coordination and uncertainty due to distribution is inherent. Time zones and local holidays do not go away. Distant communication can introduce latency and noise. Shipping and travel take time and budget. In addition, it is typical for team members to simultaneously participate in other local projects. Coordination, driven by dependencies, becomes a significant portion of total activity and demanded in uncertain patterns (that are surprising) to the teams involved. Even if one wanted to invest to adjust structures and behaviors of teams into a common standard, this new standard will not likely make sense by local norms. As Schein reminds us:

> Changing an organizations structures and processes is therefore difficult because it involves not only considerations of efficiency and effectiveness vis-à-vis the external task but also the reallocation of internal "property" [5]

8.2.3 Concurrent and Mutual Dependence

As used practically by schedulers in industry, and as captured in common project scheduling tools, dependence amongst activities is represented as precedence relationships; e.g. a task's start can begin only after another task finishes. As part of creating sequence and duration of a chain of activities, each dependence becomes a sequential relationship between a milestone in one activity with a milestone in another activity [6, 7]. The most frequently used dependencies relate the sequence of start and finish milestones [e.g. Finish to start (FS)]. If a precedent constraint is not precisely the point in time after which one milestone may follow another, a lag or delay can be characterized.

We note that the mathematical relationship describing the sequence and the lag contains very little information. **The precedence representation of dependence captures only the expected schedule consequence rather than any of the other coupled characteristics of the activities**. The underlying driver of dependence—the essential meaning—is not expressed.

Our approach models dependence as a continuous and mutual demand for coordination between teams. The satisfaction of dependencies is real and ongoing activity—coordination that matches the demand for information with the supply of information through coordination by teams. Just as teams have abilities to work, they also have abilities to coordinate, should they allocate attention so. However, an ability to coordinate is not purely determined by their own state, but as a result of their position in the total project architecture with respect to other teams. Poor coordination can trigger reduced quality, exception handling, wait, and rework. As such, the cost of coordination is driven by activity complexity, tacit knowledge, information entropy, and work culture.

8.2.4 Concurrent Dependence During Mutual Progress

As an example of concurrent dependencies, consider the Gantt chart at the top of Fig. 8.3 with two activities, one for design of drawings and the other the development of a prototype. The Gantt chart contains very little information, only the

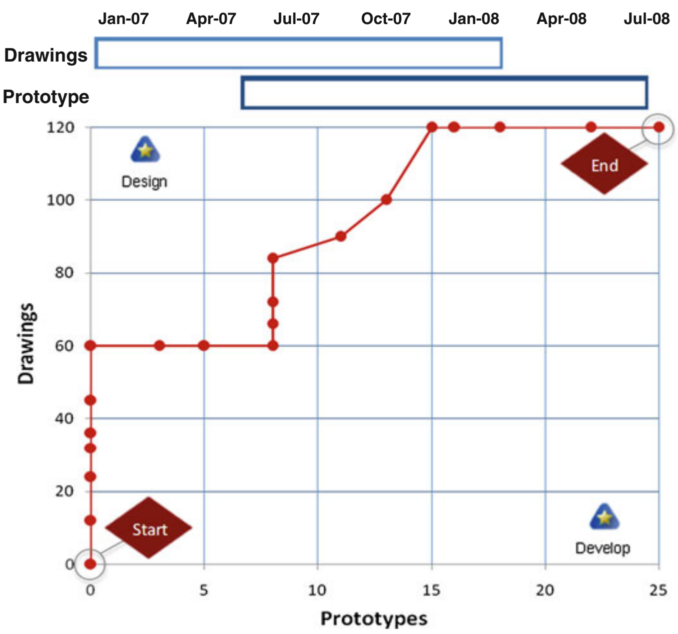


Fig. 8.3 Example mutual progress diagram

expected start and completion dates. While the overlap of the two activities on the calendar is suggestive, there are no clear insights as to any relationship between the two.

Based on the same schedule, a mutual progress diagram is shown in the lower part of the figure, with progress of Design as the upstream activity (shown on the Y axis) and progress of Develop (shown on the X axis). Rather than a shared calendar axis, we rotate the upstream activity onto a Y axis, so that we can see the mutual progress of the two activities independent of calendar. This creates a diagram which captures the mutual progress of the two activities. Units of progress—drawings and prototypes—are the measures for each axis respectively.

The expected schedule, from project start to end, appears as a red line. Each point on the line is of equal calendar time apart, in this case 1 week. Points that are closer together imply that, for that portion of the schedule, more calendar time has passed or that mutual progress has slowed. Visually one can quickly see concurrency (weeks 12–15) as the slope of the line indicates both activities making progress in the same weeks. In contrast to the Gantt chart, this diagram shows the relative and ongoing progress of the two activities. Again, even though the diagram is very suggestive (why during the first 12 weeks is there a back and forth pattern?), there is still no explicit knowledge of the dependency in the diagram.

Design information flow simulation (DiFS) was developed by Christian et al. [8] to characterize the design process as the generation and flow of information. In Christian's approach, the dependencies are depicted as areas of ongoing constraint on progress. In Fig. 8.4 on the left is a traditional precedence dependency, sometimes referred to as a FS dependence, as depicted as a concurrent dependence: the blue area completely constrains all progress until all upstream drawings have been complete. Progress in the downstream activity—triggered by its own start milestone—must wait for the full information of the upstream design activity. Importantly, one can now see beyond precedence, that the dependence is not just the relationship between two milestones, but a relationship between all information upstream with the activity downstream.

In the middle one can see a concurrent dependency that shows a strong constraint early in the downstream progress, yet as the two activities approach completion the

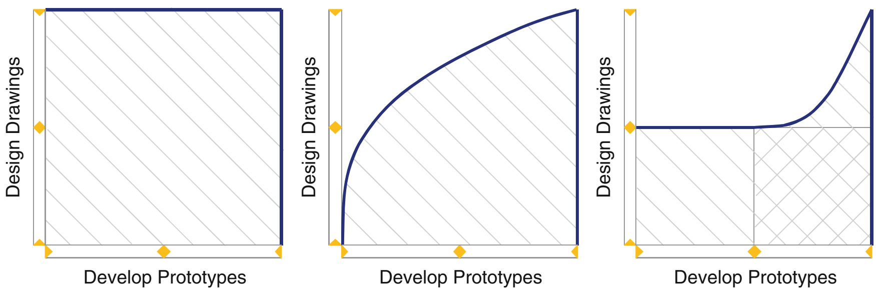


Fig. 8.4 Concurrent dependence—FS (*left*), early (*middle*), and stages (*right*)

constraint is softened to allow a pacing progress. In our method we refer to such a dependency shape as an "early" dependence: the activity is more dependent early on.

Using named milestones as markers of transition from one stage to another, a concurrent dependence can be divided ("carved") into sections with different patterns of constraint. The figure on the right shows an example of a two staged concurrent dependence: until the first upstream milestones (halfway) the dependence acts as a FS for half of the downstream scope. After that point there is very little dependence until "late" in their mutual progress.

8.2.5 Dependence as Need and Demand for Coordination

Our approach emphasizes the broad definition of *dependence as need* in contrast to precedence constraints common to existing planning methods. It is argued here that the meaning of "dependence" should go beyond a measure of sequence and be tied to an activity's own purpose for which the dependency is relevant. Thus, in our project models, **dependence is defined as need for interaction that matters—a demand for coordination—so that an activity's outcome is successful**. The characterization by Christian et al. [8] of dependence as need for information is extended in three ways:

- **Information, Results, and Attention**: a result, shared resource, or other item that is needed from the other activity, yet does not necessarily contain "information".
- **Mutual Dependence**: In complex engineering projects the activities can also be mutually dependent, and therefore not clear which activity is upstream and which downstream. Progress of the two activities is tightly coupled, with need for shared information in both directions.
- **Unsatisfied dependence triggers exception**: a violation of the constraint (a dependency not being satisfied) triggers exceptional activity, which can lead to not only wait but also errors, quality activity, exception handling and propagating consequences.

Not all coordination is perfect and timely. Realistic schedules, as well as scope, cost, and risk dimensions of project plans, should reflect not only perfection, but the chance that some things change and mistakes can be made. In turn, an organization's need to respond to oversight, decisions, change, and rework trigger new demands for coordination and work. This representation—of dependence as need—allows consideration of outcomes should the demanded coordination not be satisfied. From contingency theory, organizations follow patterns of behavior within limited rationality related to factors of organization structure and environment. Such contingency theory has been included in simulation from Cohen's 1972 Garbage Can Model to more recent *Organizational Consultant,* a rule-based expert system [9].

Another established view from organization theory is the information and communication processing role of organizations. In the 1970s Galbraith described

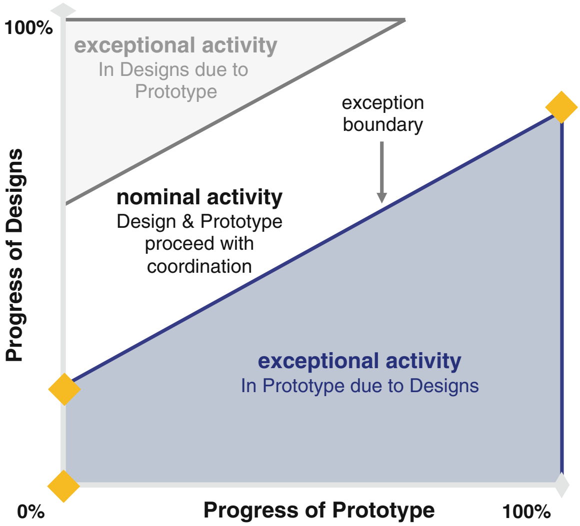


Fig. 8.5 Concurrent, mutual dependency extended beyond information flow

the exception handling behavior of organizations with established channels of communication and limits in information processing capacity. Cohen et al. have applied the exception handling behaviors in an agent based simulation called Virtual Design Team [10].

Taken together, these attributes of dependence describe a continuous need for information, results, and attention due to the progress of activities, triggering exceptional activity if the constraint boundaries are violated, and able to be depicted in both directions of dependence. Figure 8.5 shows these characteristics in one diagram. If the state of progress is in an unconstrained area, and dependence up to that point has been satisfied through coordination, then the two activities at that moment are effectively independent. The dependency is represented in both directions, from Design activity to Prototype activity and from Prototype to Designs [11]. The shaded areas indicate the general zone of exceptional activity. Thus the open area—when the two tasks can operate independently—is a zone of nominal activity. Crossing the exception boundary triggers exception handling behavior.

8.3 Analytics: Forecasts of Engineering Activity

Given a meaningful model of an engineering project's integrated product, process, and organization, we then evaluate the project through simulation. Our simulations of CE projects trace the behavior of team participants as agents. These software agents (or actors) are modeled with both work and coordination behaviors. Work behavior enables the agent to complete the skill-based scope within an agent's own domain (i.e. task). Coordination behavior allows the agent to respond and interact with other agents.

The simulator event loop begins with each agent observing the state of the project. This awareness of the environment is itself a behavior. Based on the observation, the agent selects an action to take, attempts the act which in turn impacts both the internal state of the agent and the outside environment, the project.

Participation in the project implies that there are demands on the agent, both in response to direct contract work for a task and through dependencies with work in other tasks. While specific models characterize types of demand differently, demand to work, to communicate, and to transfer results are most common. The agent makes a selection amongst these demands based on priority and availability of its own supply. The selected demand is attempted. If the selection is work which is dependent on work, information, or resources from a different agent, additional effort may be required within the event loop to coordinate across that dependency.

A key feature of the simulation is that coordination—activity to satisfy dependencies—is explicitly analyzed. *Coordination activity is real work and must be considered if realistic schedule and cost forecasts are to be generated.* Many large complex global projects have 35–45 % of real work associated with collaboration/communication. Simulation generated forecasts include the demands, feasibility, and value of coordination to overall performance. Risk due to coordination misallocation is exposed.

Typical simulation results are shown in Fig. 8.6 including "smart" Gantt charts (the chart can be queried to determine the pattern over time of an amount of work, communication, wait, etc. that is associated with team, e.g. engineers, designers, etc.). Future uncertainty is considered, including patterns of allocation attention in an unconstrained environment as well as a Monte Carlo algorithm delivering variances on key activity characteristics and team abilities. The schedule and cost produced by the simulation show a range of most likely values for that design.

In contrast to the limited information for each task in a traditional Gantt chart, the progress and exceptions in activities and the utilization of teams are visualized—not only the start and finish of these tasks and teams (Figs. 8.7, 8.8, and 8.9).

During a workshop, teams explore how each project scenario implies the project is likely to unfold. Starting exploration from a high-level architectural view of the total project, the teams then "drill down" on particular parts of the project: a product, phase, or their own combinations of work and coordination. The pace and quality of the workshop is fostered by the team's iterative ability to generate scenarios and rapidly explore their forecasts.

The continuous progress and utilization forecasts shown below provide a much more realistic insight into expected outcomes, stimulating dialogue. Teams are able to see the systemic impacts from day to day overlap of various kinds of activities, rather than idealized fixed duration and error-free tasks. They can also see combinations of their own activities with the activities of other teams, and how that combination impacts performance and systemic risk.

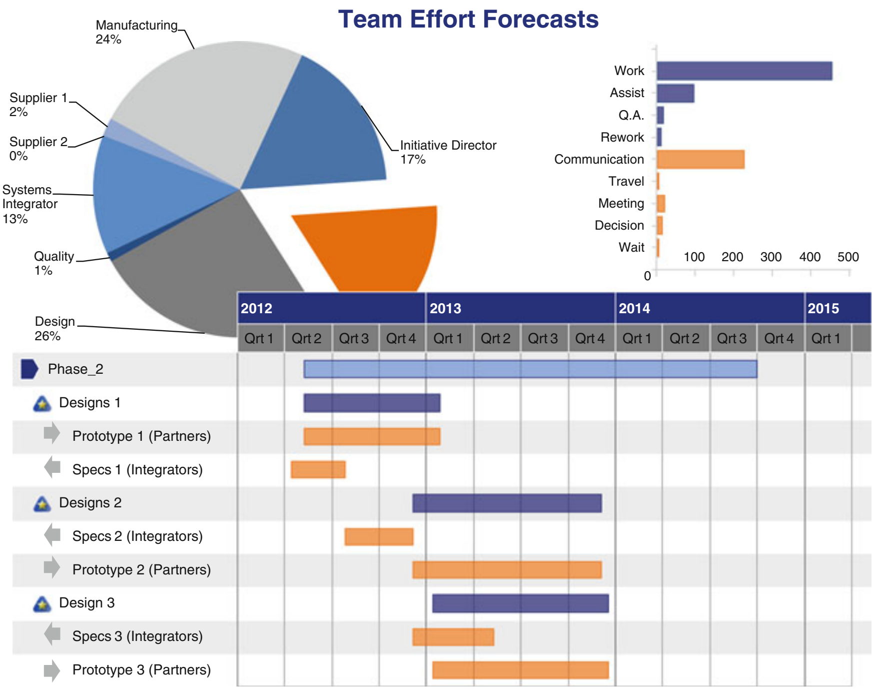


Fig. 8.6 The figure shows a typical result of a forecast. The pie chart denotes the effort in man-hours of the various teams. The pie slice representing the engineers' effort is further queried (to the right of the pie chart) as to the distribution of activity across work, assistance, QA, rework, communications, etc. A snapshot of a *Gantt* shows one team's work and coordination on the schedule

8.4 Workshops: Project Design in Collaborative Practice

Since teams joining a complex project bring embedded assumptions and practices, a critical need is to enable these teams to foresee the consequences of their own behaviors and in turn make adjustments. Models of integrated product, process, and organization combined with an analytic capability to forecast emergent outcomes sets up an engaging "humans in the loop" optimization process. This process is deployed in workshops for the cross functional team.

Design is an iterative and social process—the evaluation of choices and outcomes early-on, before committing to a course of action. By rapidly exploring possibilities—through dialogue, analysis, and prototyping—awareness is built and better results are achieved. As things change a good design is easily adjusted. Project Design is this forward-looking capability applied to engineering projects themselves. Much like design practices and tools revolutionized product development (e.g. 3D visualization, parametric modeling, QFD, CAD, CAE), Project Design can transform awareness, speed, and performance of teams on complex engineering projects.

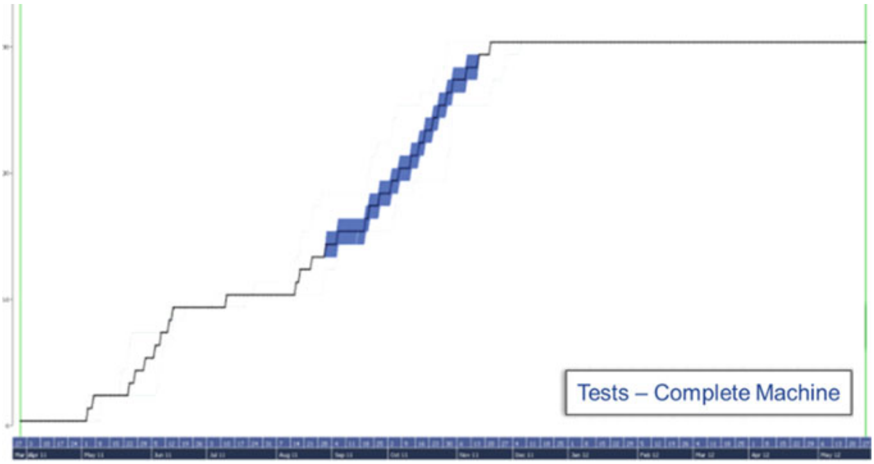


Fig. 8.7 The figure shows a **scope progress** forecast of a metric across an entire phase—in this example the tests completion for a complete machine. Other sets of tests and their most likely completion can also be compared, providing foresight into the reduction of risk based on the most critical validation. These progress timeline forecasts can be seen at various product, phase, and activity layers in the project

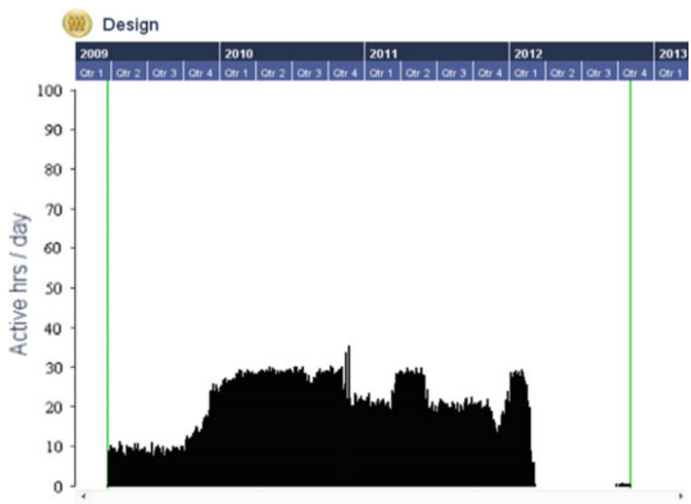


Fig. 8.8 The figure shows **total activity** for the Design team in this example project. Rather than simply an allocation assignment or capacity calendar, this forecast of expected utilization is analyzed with the interplay between various demands on this team and their skills, capacities, and priorities. Just because a team is available and assigned, does not mean it is well utilized

Fig. 8.9 The figure shows **communication** as forecast for the Design team, a subset of the total activity shown in the figure above. The demand for communication is driven by dependencies, meetings, and other factors. Still, the design team has limited capacity to respond, or perhaps other priorities. Very often, not all communication demand is satisfied, which can have delay and quality consequences

We have built a platform called *TeamPort* to support visual modeling, agent based simulation, and collaboration in Project Design (Fig. 8.10). The platform enables collaborative visual capture of the product, process, and team architectures for a proposed project. These architectures emerge similarly to traditional work breakdown structures, integrated master plans, and workflow. They differentiate themselves by a high degree of consistency and completeness that can be achieved at an agreed to and meaningful level of granularity. The platform further allows tracking of the evolution of projects, revealing the impact of any changes in key assumptions, requirements, scope, progress, uncertainty, skills, utilization, and processes.

In a workshop spanning several days, the major stakeholders and team leaders bring their skills, experience, and knowledge of the project to the table. Consistent with the fundamental practices of CE, these teams collaborate across functional boundaries to prototype their own project. The total project architecture, scope as mapped to requirements, roles and responsibilities, dependencies, and other knowledge are captured through dialogue, simulation, and iteration. The search for a desirable and feasible project plan takes time, attention, and learning. Similar to other high performance team practices, such as rehearsals in the performing arts, field exercises in the military, and practices by sports teams, the teams are iteratively exposing hidden assumptions, gaps, and converging on shared actions with situational awareness.

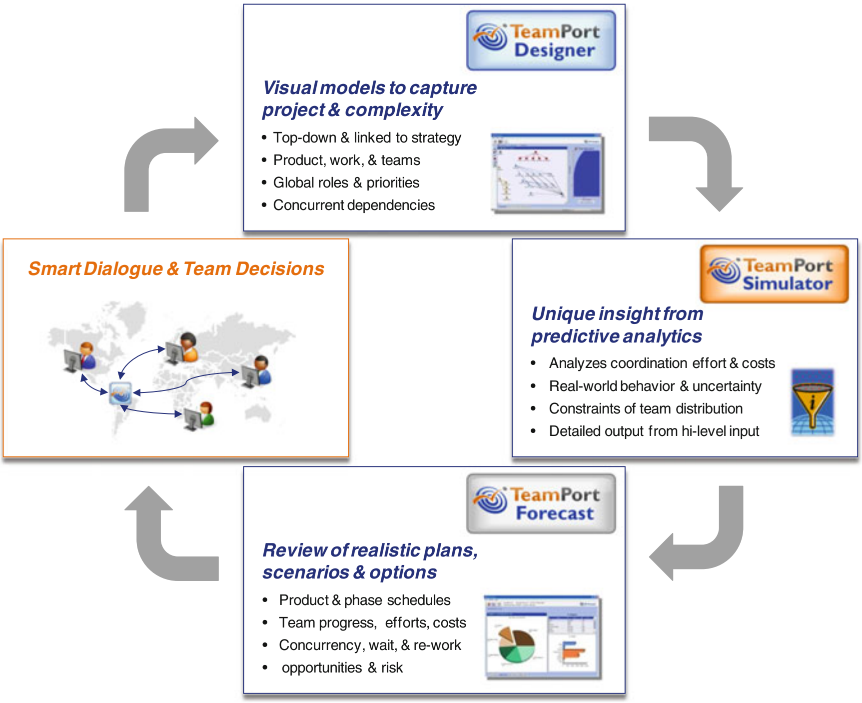


Fig. 8.10 Project design as a collaborative, iterative process

We have seen that automatic generation of an idealized schedule is not helpful if the teams are not confronted—and then agree—with the adjustments to their existing behaviors necessary to make such schedules (and cost and quality) feasible. A tension is exposed between what is hoped for and what is likely. Visual, diagnostic hints are offered during the workshops to help guide towards a Pareto "feasibility frontier", yet the real trade-offs of cost, schedule, and scope requires a social process to parallel the analytic. In that sense, the project design workshops are "Human in the Loop" optimizations, a search for root cause and correction in response to systemic and emergent characteristics of the complex project at hand. Therefore, the pace of the iteration must match and respond to the capacity of the teams to learn and adjust.

Assumptions about standard work practices may be tested, including such factors as degree of concurrency, phasing, roles, technology decomposition, system interfaces, and risk and its reduction. If embedded behaviors, in interplay with the total project architecture, lead to surprising negative or positive performance, the design of the engineering project as a sociotechnical system leads to un-learning, then awareness, and then learning of the project approaches more likely to produce positive results. The design of concurrency (e.g., where, why, how much makes sense) is specific to the nature of the system and its architecture.

The workshops are typically led by an experienced facilitator to guide the collaborative teams through the project modeling and forecast iterations. As always, a facilitator's experiences and knowledge of the particular market and technology are helpful, so that probing questions and questions of weakness in the project's design can be translated into the domain of the participants. However, the Project Design approach requires that the project leadership and teams themselves are designing; the facilitation is meant only to provide the platform and method. We have seen over time that the simple, visual, collaborative, and rapid nature of the workshops though allows teams with no background in project management nor in this particular modeling platform to engage more than sufficiently. In fact, most participants report that the Project Design process is more intuitive and engaging that the traditional detailed or ad hoc planning approaches that are displaced.

8.5 Cases

From tens of industrial cases over the last decade, two are shown in this section. The first case is new product development in industrial machinery. The project complexity was due to the inexperience of the program manager and the global distribution of the engineering and manufacturing work. The product system itself, and overall schedule, were of average complexity and scope. In contrast, the second case, from aerospace, was complex in technology risk, product system, scope, and a very unusual teaming across companies. However, the second case was within a single region.

8.5.1 First Case

A recent example for a global development program for industrial equipment is shown in below. In Fig. 8.11 a portion of the completed model is shown, with scope for testing of multiple versions of a complex vehicle manufactured and tested in several countries. Figure 8.12 shows the complete model. The complete model (including all dependencies between activities and connections between team, products and activities) can be explored. Various structured views and layers for connections can be turned on and off for clarity. As a collaborative modeling experience, teams explore parts of the project connected from separate workstations then together discuss a shared view at high resolution projected on a large wall.

Based on the visual and parametric model of the project, the cross functional team generated plans for 35 various scenarios over 2 days, balancing scope, phasing, concurrency, team roles, use of critical facilities, and risk mitigation (Fig. 8.13). Across the scenarios which captured the full project scope requested by stakeholders, the teams were able to design a baseline plan with improved likely schedule for market entry by 10 months. Importantly, the design changes and

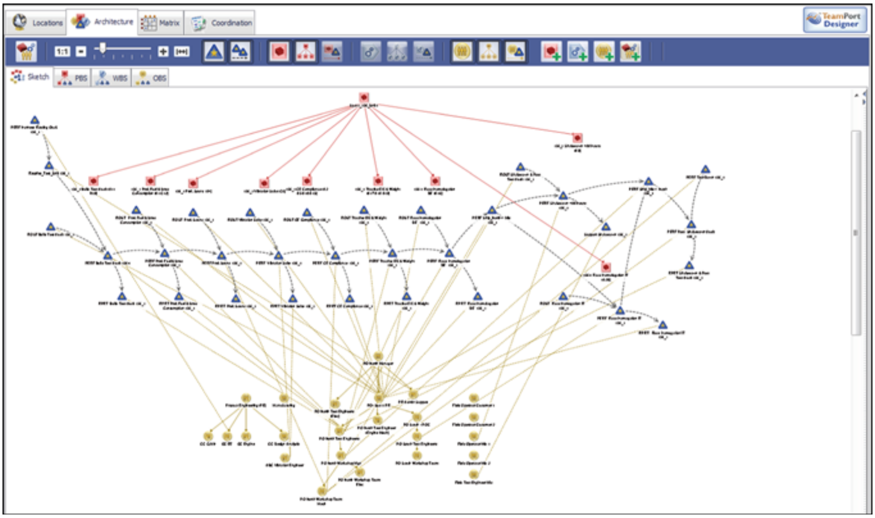


Fig. 8.11 The figure illustrates relationships at a high level between the product breakdown structure depicted by *red squares*, the work breakdown structure depicted by *blue triangles* and the organization breakdown structure depicted by *yellow circles*. Lines (*yellow*) of responsibility between and among teams, mutual and concurrent dependencies between work activities (*black dashed curves*) are shown; these can be hidden or highlighted in layers for clarity when working with the model

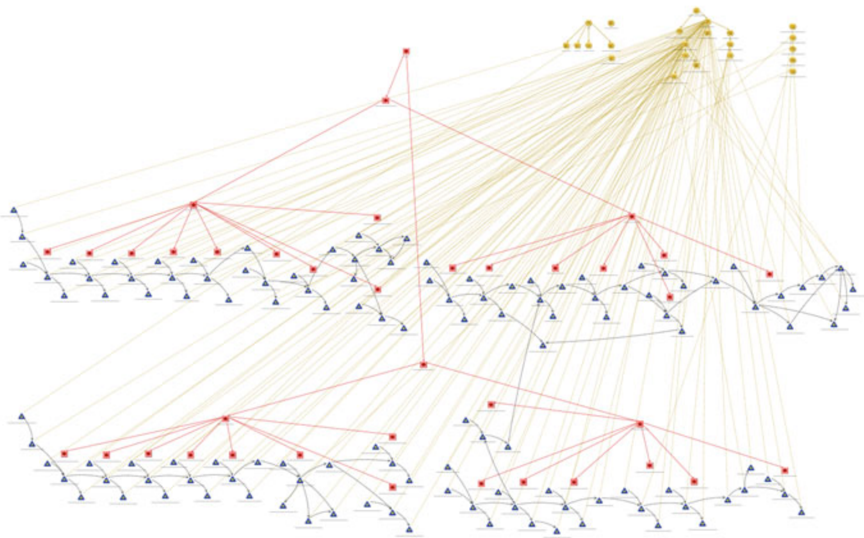


Fig. 8.12 The figure shows the entire project design model for the project described in Fig. 8.11. Projection of such a complex model onto a wall or large screen aids in working with details of the model. Additional views, as matrices, structured layouts, and lists are also generated to promote exploration and a "forest for the trees" view of project architecture

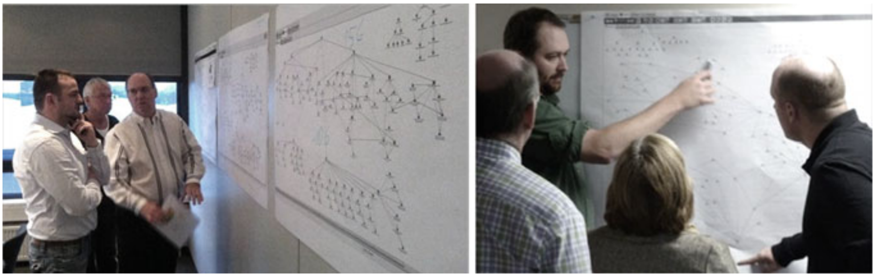


Fig. 8.13 The figure shows workshop design sessions where a printout of a project design model has been posted on the workshop wall. The model in the figure on the *left* is the one shown earlier. Whether on paper or through real-time visual interaction, the teams quickly come to form a share mental model of their complex work and their own role in it

trade-offs to achieve this 10 month gain were proposed and made together, while balancing systemic impacts on cost and risk.

In a matter of days, the cross-functional team was able to generate a meaningful baseline plan amongst alternatives. The participants were able to easily grasp and interact with the project model with no method training, focus on portions while quickly stepping back to consider total architecture and expected outcomes. In contrast to the detailed Gantt Charts and ad hoc spreadsheets employed previously, the quality of engagement and robustness of the resulting plan were much improved.

8.5.2 Second Case

The next case involves two companies, A and B that formed a joint venture to develop a new product. Normally A and B are competitors. Although the venture had been running for several months before the Project Design session, we mutually decided with management to apply the full Project Design methodology ab initio. When we arrived on the scene, the program was midway through the concept development phase and had experienced cost growth from initial estimates.

In the workshop discovery process, ethnographic interviews of key program contributors revealed a stark difference in the new product development cultures of the two partners. For concept development, Company A used a separate team and a flexible, rapid prototyping and learning process compared with Company A's more exacting process for detailed design and development. On the other hand, Company B's culture was to apply its rigorous, detailed design and development process to concept development. In consequence Company A was almost always waiting for Company B to finish its activities so that their work could be integrated and tested. It also transpired that both companies, and, indeed, some groups within a company, used different project estimating methods and earned value management measures (e.g., level of effort, percent of milestone completion, actual milestone completion). All told there were four project management tools in use across the two companies: Excel, MS Project, Primavera and SAP.

The initial Project Design model of the program, drafted by a few key players with facilitation help, did not close with the venture's current estimates of schedule and cost. The sense was that Project Design had overestimated the "Partnership Drag" or coordination effort and that the model contained incorrect assumptions about Company A, which had not chosen to participate in the initial modeling session. Both companies thought that coordination effort was already built into the reference programs that each used as the basis of estimating their parts of the current program. In view of delays waiting for Company B to complete its activities, Company A conceded that "Partnership Drag" might account for an additional 10 % of coordination time that was not included in the original estimate. As the program execution unfolded, it also came to light that Company A had adopted an inappropriate reference program for its estimating process.

Most importantly, during initial modeling, we came to realize that the original program plan was success oriented, that is, there were no provisions for risk mitigation included. About 15 % of the cost growth was attributed to "discoveries and new knowledge," a byproduct of unforeseen uncertainties and emergent behavior.

Company A's culture for conceptual design could be characterized as "hands-off elegance," which translates as hubris. Its venture management finally agreed to meet with the Project Design facilitators to straighten out the faulty assumptions in the program design model that evolved as the product of a three-day workshop (without the participation of Company A). After this was accomplished the venture management team decreed that there would be an additional, fourth day to the workshop with mandatory attendance of key contributors from both companies. The fourth day of the workshop produced tens of feasible plan forecasts, a subset of which came close to the desired parameters. Several options were identified, which needed further study, for shortening the schedule and reducing the cost of the program.

In another example of reversing a poor cultural habit, the Project Design methodology brought new understanding about the program to the several members of the program team who were not part of the original planning and were, in fact, looked upon as cogs in the program's "machinery." This improvement occurred as part of the social process associated with the design workshop.

8.6 Comparison with Contemporary Approaches

Along with a unique analytic treatment of coordination activity, the Project Design concept is built on our field experiences with ideas pulled from different lines of contemporary thinking, practices, and tools. We focused on improvements in the areas of representation, analytics of prediction, and social process. Why are these improvements necessary? Because the established body of knowledge and standards of program management, although satisfactory for stable environments, aren't reliable in today's environments that are expanding in both technical and social complexity in ways that exhibit emergent behavior. Today's program teams need to be able to anticipate and react quickly to surprises that grow from seemingly insignificant anomalies.

Better representation of interdependencies among elements in the program architecture together with analytics for scenario evaluation are two key improvement areas. In programs encumbered with "natural" variation and other uncertainties that are foreseeable but likely ignored, as well as with the unforeseeable consequences of the choices that teams have to make, the probability of surprising outcomes is high. These factors form the technical dimension of program management. How teams make choices, or which demands teams choose to satisfy or ignore, is conditioned by deeply-seated habits and culture. To tackle this dimension, we seek to improve the social process of program management; we call this social process "Human-in-the-Loop Optimization." Project Design is above all a learning process that continues to occupy our research.

The architectural representation of a program as the three elements of product, process and organization is imbedded in the Quality Function Deployment methodology that started to be used in the 1980s in the United States; the three-part representation was also a feature of the master database, called the PPO, that was one foundation of DARPA's initiative in concurrent engineering (DICE) conducted in the late 1980s to early 1990s by a team of industries and universities. The Project Design improvement, in this case, has been to employ a graphical engine to simplify building and manipulating the architectural representation and to make visible the interdependencies and dependencies among its parts. Through our notion of scope, Project Design offers a convenient way of capturing and displaying interdependencies and graphically tracking progress in resolving them. These and other reports create situational awareness among the performing teams.

The analytics in Project Design are associated with an agent behavior based engine that runs a parallel large number of "demand-activity-supply" loops, perturbed to emulate variation for each program design configuration. In essence, each loop represents a collision between demand and supply that is mitigated with repeated iterations. Within the simulation engine, each such loop accounts for demand work, rework, and their complexity, coordination demand and constraint, roles and responsibility, and availability and skill level of resources. Because of its approaches to representation and simulation, Project Design sidesteps the traditional project management network methodologies such as program evaluation and review technique (PERT), critical path method (CPM) and conditional diagramming method (CDM). It is not necessary to identify either the critical path or the critical chain, which Goldratt persuasively argues is a more important measure of program schedule than the critical path [12]. Instead, Project Design automatically identifies the program activities and interdependencies that are currently the largest levers on overall program performance and, therefore, should attract the greatest attention of the program teams. As an input option to the Project Design system, one can specify to ignore (or assume) coordination activity; the resulting simulation then approximates the CPM solution.

The social process of Project Design is presently still evolving along with our ethnographic and analytic research of each workshop event and surrounding cultural experiences. In Chap. 3 we discussed the socio-cultural model of an organization and we recounted Gharajedaghi's [13] notion of culture as operating system. A motivating question, recently put to us by a physicist, is, "Can culture be modeled?" We humbly believe that the answer is affirmative, although the confirmation will take more work and will be a topic for another day. Moreover we assert that traditional models of work indeed represent cultural characteristics, though hidden as embedded assumptions and simplifications.

For now we are concerned about the social and cultural interactions, within and between teams that attend the collaborations necessary to resolve interdependencies. For example, in researching collaboration Schrage has recognized that managing relationships is more important than managing individuals [14]. Although Schrage's thesis of "No More Teams" finds that teams are neither required nor necessarily desirable for collaboration in today's environment, research on high-performance teaming, some of it dating to World War II, has long supported that the best teams strike a balance between task orientation and attention to relationships. Whether in a team setting or not, collaboration is unlikely to occur between two experts who will not share knowledge with one another because of a soured relationship. It is up to the Project Design workshop facilitator to establish norms of behavior and ground rules that will build relationships. We also note that Project Design workshop participants are often leaders or representatives from performing teams and may not be members of the same team. It is, therefore, all the more important that the Project Design workshop sets and demonstrates the tone for the culture of the entire program.

Schrage further argues, with several examples, that the social process of collaboration is considerably improved by the use of collaboration tools. We identify Project Design workshops as our fundamental collaboration tool, because the workshop gathers together contributors from many corners of an organization (or organizations) and encourages them to create productive relationships (collaborations) to design a feasible program. The software support system of Project Design is another collaboration tool; since it rapidly executes simulations of different program design scenarios, it provides the workshop participants with near-instantaneous feedback on the feasibility of their designs. This feature brings three benefits: (1) collaboration can proceed at the speed of human conversation; (2) collaborators can learn rapidly about the behavior of their system (program) design choices and master their optimization; and (3) collaborators can feel a sense of significant accomplishment for a relatively small investment of their time. In principle, given the architectural representation and simulation analytics of Project Design, the technology exists to automate the optimization of a program's design. However, automating the optimization would vitiate the relationship building, deep learning and feelings of accomplishment and ownership that derive from the current, human-in-the-loop social process of Project Design.

In 1997 the National Center for Manufacturing Sciences (NCMS) initiated a benchmarking investigation of global companies to discover exemplary product development processes. After the benchmarking team was exposed to Toyota, it stopped looking at other companies and redirected the investigation to study Toyota's product development process at a deeper level. The results of this study were documented by Michael Kennedy in the book Product Development in the Lean Enterprise [15]. Toyota had created a knowledge-based development environment (culture) that rested on the knowledge of individual workers: their understanding of needs, information availability, responsibility and teaming interaction. In this knowledge based environment, the system architecture emerges from the interaction of all functional perspectives. Interaction occurs through natural communication and through integrating events where team decisions about next actions are taken. The parallels with the learning, interaction (collaboration) and integrating events (workshops) of Project Design are evident.

8.7 Benefits

Engineering project plans created through a Project Design process are generated more quickly with increased accuracy by incorporating realistic drivers of feasibility. Although Project Design can handle a large amount of detail, teams soon discover that an elevated level of abstraction provides better and faster learning about essential program and system knowledge than a very detailed plan. Besides work demand and ability variation, Project Design allows teams to account for foreseen uncertainties by building risk mitigation activities into the program's process structure. When program teams are surprised by an unforeseen or emergent uncertainty, they can rapidly incorporate learning loops and recovery actions into the program design and assess the extent of any setback on overall schedule and cost by running simulations.

We adopt the idea of culture as operating system and help the teams to build a common culture in the process of designing and executing the program. This step is essential, because most global teams start with their individual cultures and, therefore, are a recipe for social complexity [16]. Project Design begins with ethnographic interviews, following a tested format, of a number of key team members. From the interview, facilitators identify cultural issues or behaviors that may need to be dissolved or replaced, either during the design of the program or during its execution. The data from each new workshop feeds ongoing research to improve the social process of Project Design.

Another benefit is found in the adaptability to either external or internal changes. If a customer orders a change, if something in the external or internal environment changes in a way that invalidates an assumption, of if a risk mitigation activity fails, to name three circumstances, the design of the project can be modified rapidly, enabling the teams to re-design the program with accuracy and awareness [17].

8.8 Conclusion

Project Design is a platform for effective planning and ongoing dialogue of teams on modern, complex projects. These projects often have no feasible central planner with complete awareness of work practices, background knowledge, and priorities of dispersed teams. Even if a central planner or automated scheduler possessed the information required for a good plan, there remains need for a social process during which teams take time to propose, negotiate, prototype, iterate, and learn. If deeply embedded assumptions—hidden and effective in past projects—are to be exposed, the project design process must predict unexpected outcomes at a pace of learning by teams. In order for teams to deviate from their existing work culture, unexpected impacts from misallocated, poorly timed, or missed coordination need to be confronted as surprises.

A very good workshop session often begins with forecasts that cause great concern. In this way the *Project Design* process enables teams who might otherwise not share work culture to adjust, develop shared awareness, and converge towards a common, feasible, and optimal plan. In the cases where disparate behaviors and abilities lead to challenges that cannot be overcome within the horizon of the project, the project architecture can be otherwise designed to mitigate potential negative impacts.

Stability of experience in a learning environment can turn complex activities into simple ones as behaviors of the system are understood—people develop shared work practices and build up tacit knowledge. Learning, as the generation and transfer of information across dependencies, drives transformation from complexity to simplicity.

In contrast, a traditional master planning or automatic scheduling method is not likely to lead teams to explore embedded local behaviors which drive misallocated, poorly timed, or missed coordination. In this way the *Project Design* process enables teams who might otherwise not share background to develop awareness, adjust, and converge towards a common, feasible, and optimal plan. In the cases where disparate behaviors and abilities lead to challenges that cannot be overcome within the horizon of the project, the architecture can be designed to mitigate potential negative impacts.

Case studies over the last decade show that small architectural changes may lead to surprising outcomes as projects become more complex. From each team's perspective, the way that the integrated architecture generates demand for work and coordination may appear in combinations inconsistent with the team's existing work culture. The design of a project may unknowingly disrupt the potential of embedded practices, abilities, and knowledge. If a team's work culture acts as an organizing driver to decrease uncertainty in the integrated sociotechnical system over time, a surprising sudden shift in various demands and costs of coordination will increase uncertainty. In complex projects a small change to alignment of the team's abilities (supply) to the need for work and coordination (demands) can lead these very same embedded practices to be wasted or, moreover, trigger unexpected delay, poor quality, and propagating rework. A team unaware of these impacts—following their own best judgment—may in fact be a cause of systemic poor performance. In these cases, given the counterintuitive root cause of these difficulties, teams in frustration may harden their belief, instead assuming that the cause of difficulty must be the behaviors of other teams.

Project Design is part of a new generation of methods that seek to model real-world dynamics of project work, provide early, architectural views of the project as a sociotechnical system, and allow forecasting of the range of likely outcomes in cost, schedule, and quality. These methods—rather than displacing people during the planning process through automation—leveraging interactive visualization and collaboration technologies to involve people in exploring the range of structures and behaviors. The design and optimization of the integrated project as socio-technical system includes the awareness and commitments of the people who will perform together.

References

1. Moser B, Mori K, Suzuki H, Kimura F (1997) Global product development based on activity models with coordination distance features. In: Proceedings of the 29th international seminar on manufacturing systems, Osaka, pp 161–166
2. Bucciarelli L (1994) Designing engineers. The MIT Press, Cambridge
3. Holt A (1989) Organizing computer use in the context of networks. In: COMPCON Spring'89. Thirty-fourth IEEE computer society international conference: intellectual leverage, digest of papers, pp 201–207
4. Oravec J (1996) Virtual individuals, virtual groups, vol 11. Cambridge University Press, Cambridge
5. Schein E (2006) Organizational culture and leadership, vol 356. Wiley, Hoboken
6. Kerzner H (2013) Project management: a systems approach to planning, scheduling, and controlling. Wiley, Hoboken
7. Mantel S, Meredith J, Shafer S, Sutton M (2011) Project management in practice, 4th edn. Wiley, Hoboken
8. Christian A, Grasso K, Seering W (1996) Validation studies of an information-flow model of design. In: Proceedings of the 1996 ASME design engineering technical conferences
9. Baligh H, Burton R, Obel B (1996) Organizational consultant: creating a useable theory for organizational design. Manage Sci 42(12):1648–1662
10. Jin Y, Levitt R (1996) The virtual design team: a computational model of project organizations. Comput Math Organ Theor 2(3):171–195
11. Ni M, Luh P, Moser B (2008) An optimization-based approach for design project scheduling. IEEE Trans Autom Sci Eng 5(3):394–406
12. Goldratt EM (1997) Critical chain. North River Press, Great Barrington
13. Gharajedaghi J (2011) Systems thinking: managing chaos and complexity, 3rd edn. Elsevier, Burlington
14. Schrage M (1995) No more teams: mastering the dynamics of creative collaboration. Currency Doubleday, New York
15. Kennedy M, Ward A (2003) Product development for the lean enterprise. Oaklea Press, Richmond

16. Kahane A (2004) Solving tough problems: an open way of talking, listening, and creating new realities. Berrett-Koehler, San Francisco
17. Moser BR, Wood RT, Hiekata K (2014) Risk management in the design of engineering as sociotechnical systems. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc international conference on concurrent engineering, IOS Press, Amsterdam, pp 635–646

Chapter 9
Systems Engineering

Alain Biahmou

Abstract Unlike the first cars, which essentially have been mechanical systems, nowadays cars have become very complex mechatronic systems that integrate subsystems created in a synergy between people from different domains such as mechanical engineering, software engineering and electric and electronics (E/E). This fact has increased product complexity in the last decades and therefore the product development complexity. Complexity is multidimensional and consists of product, process, organizational, market as well as use complexity. A methodology for mastering complexity is Systems Engineering, which actually means applying systems thinking to tackle the challenges of creating complex products. The focus of this chapter is providing a deep understanding of systems engineering (SE) as well as a rough recommendation for companies that might be interested in implementing SE. Thus concepts for implementation are proposed. As an entry point, the context of product creation is presented with the challenges that are linked to. The need of appropriate methods is emphasized and the application of SE is motivated. In order to present SE as it is applied in the practice, SE processes are described in detail and the artifacts of the different steps are highlighted. For performing the processes described, SE tools and methods are presented. The important role that the company organization and the project management both play for SE projects as well as SE success factors are highlighted. Additionally, a proposal for an introduction process for SE is elaborated. A selection of functional features that can provide a cutting-edge advantage when practicing SE are presented and discussed. Two case studies are illustrated in order to provide real applications of SE and therefore an additional orientation for SE implementation. The relation between SE and Concurrent Engineering is addressed and some future challenges of SE are identified.

Keywords Systems engineering · Systems thinking · Mechatronics · Complexity management · V model · RFLP approach

A. Biahmou (✉)
EDAG Engineering AG, Reesbergstr. 1, 36039 Fulda, Germany
e-mail: alain.biahmou@edag.de

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_9

9.1 Introduction

The challenges regarding complexity have increased in the automotive industry, in the aerospace industry and in the most relevant areas of engineering during the last decades. The complexity can be observed regarding the product (technical complexity) as well as the processes of its creation (organisational complexity).

Product complexity is characterized by factors such as increasing of functions and components, but also component complexity, when a component integrates more functions and subcomponents. For instance, the number of electric components has increased in cars and therefore, the total length of the electric cables used in today's generation of well-known cars has reached a multiple of their initial values. Furthermore, components such as headlamps have been improved to include more functions, but also more sub-components such as sensors which are connected with the CAN-bus. These very small examples can give a brief impression of the real challenge that is to be tackled in product development nowadays and in particular, in the automotive industry.

Besides, complexity also is enhanced by the dependencies that must be covered during the product life. These dependencies may be existing between the components of the products as well as between these components and external actors such as project members, external documents and so on.

Furthermore, considering and integrating new technologies in existing products also is a source of complexity. A closer look on the consumer market reveals what can happen when a company producing cameras ignores advances in digital technologies, or when a company designing cell phones does not pay attention to the technologies that led to next generation devices, the smartphones.

Process complexity includes among others the methodical and collaborative procedures which are performed for product creation, for instance component integration. Complex products are developed by diverse disciplines—e.g. mechanical engineers, electric/electronic engineers, software engineers—using different processes, methods, tools and especially vocabularies. It is important for all these disciplines to understand the system as well as each other in order to conduct the project to success. This requires a meta-model of the system as basis for the work. Based on the system meta-model, a product model that integrates the partial models can be created. A partial model represents the perspective of an arbitrary discipline on the product model.

In practice, a challenge consists of creating such a product model, identifying explicit and implicit relationships resp. dependencies between the existing different partial models (see Fig. 9.1). Generating the partial models and updating the product model in the right sequence after modifications have occurred, is an additional challenge. In this case, investigating change propagations and taking them into account may be essential for being able to generate the right models.

Experience has shown that communication in projects as well as accessing data of other disciplines are important tasks that need more efficiency. This is due to the fact that the disciplines that own the data commonly use different data formats as

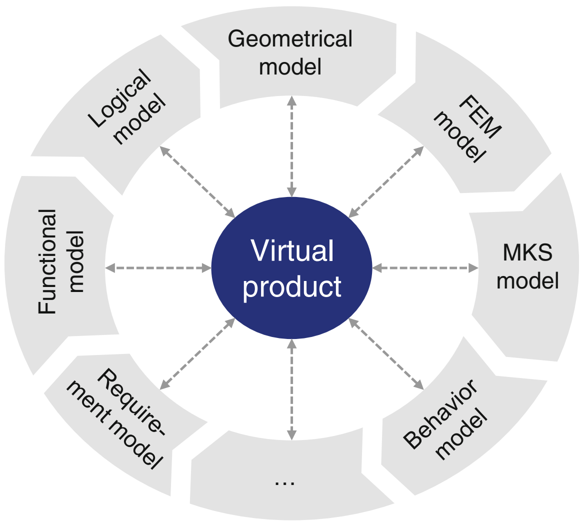


Fig. 9.1 Different views of a product

well as data management tools, for which access is not granted for data consumer (s). A PLM study has confirmed communication and data search as a very time consuming activity during product development (see Fig. 9.2) [1].

Another factor in car development is the extended enterprise. A high amount of companies of different types (see Fig. 18.2) are involved in the development of a vehicle. These companies generally are located in different countries and continents, since most of OEMs are global players, that is, modules or derivatives of vehicles often are developed all over the world (see Chap. 7). Thus, consulting services and engineering services providers as well as module suppliers that are

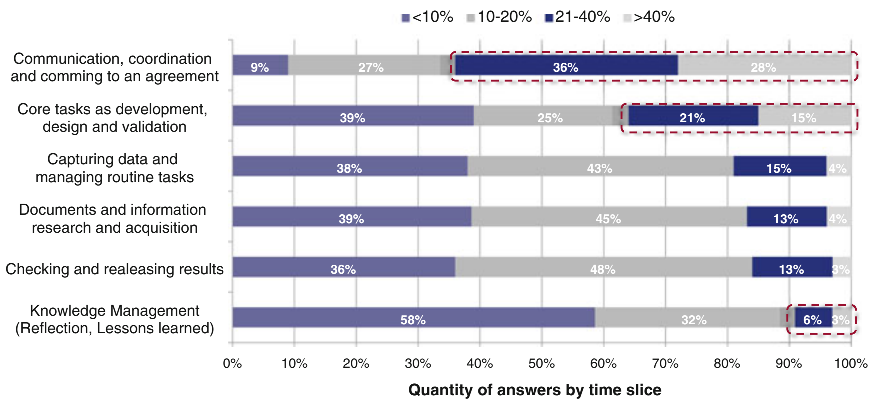


Fig. 9.2 Complexity in product development/time consuming tasks [1]

involved in the vehicle development process need product data in order to perform diverse tasks, e.g. Digital Mock-Up analyses [2]. These actors need also to actively participate to the product creation by interacting with project teams sometimes using different processes, methods and tools.

The interaction of the companies involved in the development of cars enforces therefore an exchange of data (e.g. CAD data) between different parties. The data exchange includes not only a file transfer between the OEMs (original equipment manufacturers) and the first tier suppliers, but also the suppliers with their own sub-contractors. Figure 18.2 shows a sample data flow within a network of companies that are collaborating for a project. Managing a project which is conducted in such a widespread network leads to answering questions about data security, once more enhancing the technical and organizational complexity [3].

Apart from engineering issues, Winzer presents globalization as an additional complexity factor, since it leads to an explosion of the number of stakeholders, the number of laws and homologation as well as country specific customer interests to be considered. Globalization can lead to a higher number of suppliers [4]. Mass customization also is a further factor which increases complexity as it leads to an increasing of individual functions [4]. Mass customization helps providing individual products to customers (see Chap. 14). This yields the creation of a higher number of design alternatives, that complicates not only the product creation in term of simulations, data management and configuration management, but also the after-sale phase when it comes to deliver services, e.g. providing spare parts. A further complexity factor is the miniaturization, due to the fact that a general trend is towards smaller and compacts products [4]. The miniaturization of systems yields a new adjustment of system components, but may lead also to adopting new and complex manufacturing processes, e.g. micro machining. To sum up, aspects of complexity are not limited to market, product complexity, organizational complexity and process complexity (see Fig. 9.3) [5].

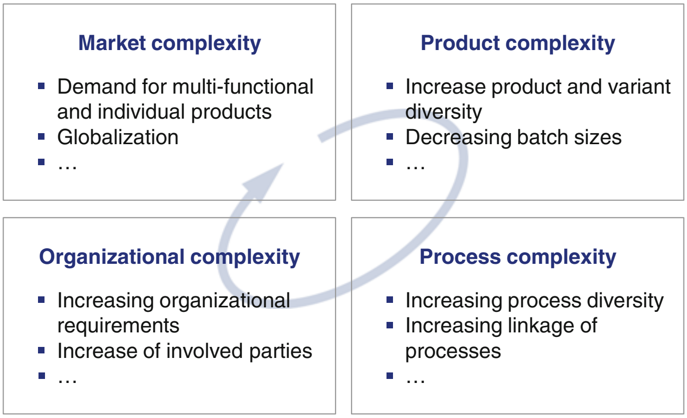


Fig. 9.3 Types of complexity [5]

The description of the complexity which has been presented above emphasizes the need of methods for mastering complexity in product development. The content of this chapter follows this premise. In Sect. 9.2 the motivation for systems engineering (SE) is highlighted. Section 9.3 describes SE in practical use (tools and methods, organization and project management, architecture). Section 9.4 draws a concept for Functional Blocks to support SE. In Sect. 9.5 introduction of SE in a company is discussed. Section 9.6 shows case studies, followed by discussion in Sect. 9.7.

9.2 Motivation for Systems Engineering

A proposal for complexity management has been presented by Schuh and Schwenk [6] who proposed a reduction of variants, but this may be problematic since the carmaker should decide in that case not to fulfil certain customer requirements. This could be interpreted as a lack of flexibility, especially if competitors provide appropriate products.

An established methodology for mastering complexity is SE, which actually means applying systems thinking to tackle the challenges of product creation. Thus what do system and systems thinking mean? Many definitions of the term system already have been provided by research works. A system can be apprehended as a set of components which are linked by relations, forming a whole. Hitchins defines a system as an open set of complementary, interacting parts with properties, capabilities, and behaviors emerging both from the parts and from their interactions [7]. Further authors make a difference between systems, systems of systems (which are built by components that are large-scale systems), mega systems and intelligence-based systems which are able to comprehend, understand and profit from experience in order to adapt to changes of their environment [8]. A system is made up by the complex networking of resources such as manpower, equipment, facility, material, software, hardware and so on. Resources are to be considered as subsystems which interact with each other within or beyond the surrounding system. A system is characterized by inputs, outputs, internal processes and mechanisms as well as constraints [9].

Systems thinking is a pre-requisite for applying SE since the multidisciplinary teams involved in product creation have to understand the system-of-interest as a whole. Collaborative systems thinking is an emergent behaviour of teams resulting from the interactions of team members and utilising a variety of thinking styles, design processes, tools, and communication media to consider systems attributes, interrelationships, context and dynamics towards executing systems design [10]. System thinking is not new in the product creation area. Design methodologies that have been presented in the past that can be applied for different areas beyond mechanical engineering have highlighted systems thinking [11–13].

Lindemann also has presented an approach to complexity management, which highlights the connectivity of elements—product components, people, documents-involved in product design. Further methods for structural complexity management mentioned are systems dynamics, operational research etc. [5]. Although diverse methods address the complexity management, that topic still remains a main challenge in the industry: this fact is a matter of evidence when thinking about some car makers who are recalling their cars in order to fix failures [14].

According to the International Council on Systems Engineering (INCOSE), "SE is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem. It integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept over production to operation. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" [15].

SE focusses on the lifecycle of technical systems, that lasts from the first idea to the recycling. It apprehends the system to be developed as a whole artifact, which interacts with its environment. Moreover, SE takes the relations between the system and its components into account and its main purpose is coordinating the disciplines involved in product development [4]. This apprehension is very close to the approach of Hitchins, that emphasizes the fact that a system is active, interactive and adaptable to its environment. Further key aspects of Hitchins' approach are the human factor on the system, its dynamic context, its synthesis, holistic character as well as its analogy to an organism [16].

SE includes management and engineering and considers that a system is more than the sum of its elements. It is a continuous iterative process which includes multidisciplinary teams and is applied throughout the product lifecycle [17]. Although SE has developed itself very quickly in the last decades, there are approaches to improve the traditional methods and, therefore, the quality of systems. Tolk et al. [8] have characterized intelligence-based systems while emphasizing the role of semantics, simulation and agents. Works presenting computational intelligence in the meaning of automated or semi-automated processes for SE have been presented in the past [18–21].

Intelligence-based SE can be applied for developing different complex products such as nuclear plants, airplanes, cars, space shuttles, machines. It helps conducting simulation already in early development phases in order to reduce costs and improve quality. It can be applied also to automatically generate hardware or software specification. Especially in the automotive sector, it has contributed to tackle an important number of challenges going from monitoring driver inattention, enhancing pedestrian safety to the control of mobility systems [22–27].

SE also has emerged from a general approach to a more and more specific approach for a couple of domains such as automotive, aerospace, medical and manufacturing. A further SE area that is being established is Product-Service-SE [28, 29]. The motivation of this evolution is explained by specific branches of SE being more efficient to solve domain relevant problems. While the most known publications about SE deal with the development of a specific product (e.g. a airplane, car) and therefore, are focused on technical aspects, the future approaches will be broader.

Some perspectives may be the application of SE in order to determine how engineering systems can interact with other systems taking social, economic and environmental factors into account. SE could be important for tackling challenges in the areas of critical infrastructure, health care, energy, environment, information security and other global issues [30]. Upcoming challenges of SE will be discussed in Sect. 9.7.

To sum up, complexity management as well as lacks in today's product development methods (e.g. requirement engineering) make SE necessary. SE targets a cost reduction by implementing methods to ensure a good quality design, a traceability between components and processes, that can help getting a better understanding of the system and therefore managing the systems development. In order to apply the SE approach, systems thinking capability, tools as well as methods are necessary. It is crucial to define system boundaries, interfaces as well as inputs and outputs of the system.

9.3 Systems Engineering in Practice

The main objective during systems development is the achievement of stakeholders concerns, who may be customers, owners, vendors or any person being related to that system. This is done by designing and integrating methods and models within the system, but also with other systems. Integrating means to network the break down structure of the different sub-systems, components and processes involved. Based on this, verification and validation which are two major tasks are to be performed in order to realize the quality standards [9].

Figure 9.4 shows two simplified schemas of the SE process, which have been proposed by Dikerson and Mavris (right) and Pineda and Smith (left) [31, 32]. This shows that even in the SE community, standardization remains a challenge. In an attempt to reach a common understanding, Ryschkewitsch et al. [33] has presented a set of definitions related to SE and its actors.

Many modified schemas already have been presented based on the V model in different variations. It is about an iterative and holistic process. For sake of brevity, the different phases will not be detailed. Practical recommendation to requirement engineering and management is provided by appropriate literature [34, 35], functional analysis in SE addresses the transformations to be performed in order to obtained required systems outputs from available inputs [36]. Elements of functional analysis have been presented among others by Buede [36] and systems integration [37, 38], verification and validation as well as testing also have been addressed by a multitude of scientific works [39].

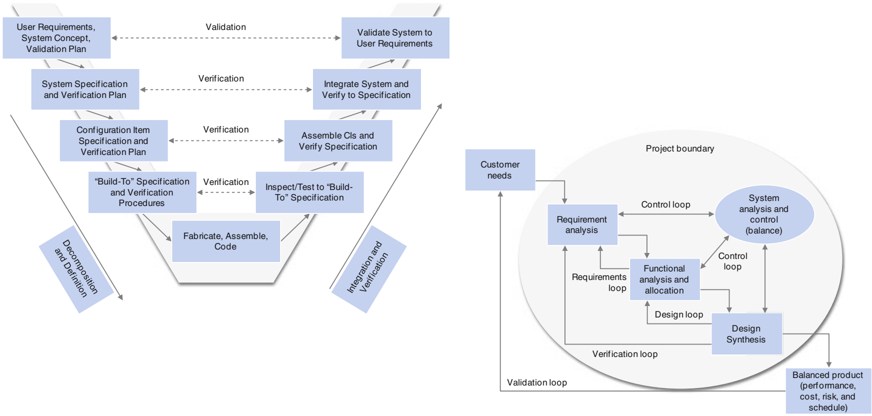


Fig. 9.4 Systems engineering process [31, 32]

From a SE point of view, requirement tracing is a premise, that means it should be possible to trace the impact of each requirement on the functional, logical, physical model as well as on verification and validation. Therefore, analyses can be performed in order to investigate to which degree a requirement has been fulfilled. Requirements engineering includes the generation, formalization, decomposition, analysis and management of product and service requirements with the objective to verify and validate them (see Chap. 5). Decomposing requirements is important since they often are formulated by persons (e.g. customers) who are not familiar with requirements engineering.

Based on requirements engineering, system functions are derived in order to create a functional model. Functional analysis consists of formalizing and decomposing system functions that are to be realized later by the design [32]. The next step consists of identifying functions that belong to the same sub-system, it is a phase during which subsystems are identified from the designed functions. Therefore, the output of this activities are groups of functions. Moreover, it is important to notice that the identified sub-systems can complement the use-cases that have been defined at project start, since the initial use cases are merely focused on fulfilling stake holder requirements. However, supporting functions and use-cases are necessary to get the whole system running.

Thus, test cases are to be defined in order to identify further inputs for the creation of the whole logical system, that enables all system components to operate together logically. Based on the groups of functions resp. sub-systems identified, a logical model of the whole system is created and behavioral models are created or generated and attached to its components. The simulation of the logical system is the basis for the creation of the overall system concept, that provides inputs for the physical design of the system in the different disciplines involved. Therefore, the disciplines can perform a concurrent development process based on system-oriented-concepts as well as a common model (see Sect. 9.3.1).

It is important to remark upon the requirements loop between the Requirements analysis and Functional analysis as well as the Design loop between Functional analysis and Design synthesis. Both ensure that the requirements analysis, the functional analysis as well as the design are continuously examined and updated. A verification loop insures that the tasks are done right, whereby a validation loop has the objective to check that the right things have been done, that is, an end product in which performance, costs, risks and scheduled characteristics are successfully balanced to satisfy customer expectations. The verification includes semantic, syntactic and plausibility checks.

The validation has to check how far the systems model corresponds to the real world. The tasks regarding systems analysis and control consist of overseeing and controlling all phases and activities of the System Engineering process [32]. Validation is crucial to ensure that the main goal has been achieved. Kolonay has introduced the concept of physics-based models to validate the conclusions made for concepts and system specification [40]. A further interesting approach consists in defining value ranges to reduce eventual gaps between the reference values and the actual values of component interfaces as well as tolerances between energy, material and signal flow [41].

In other articles, the so called RFLP approach, that stands for Requirements, Functional, Logical and Physical modeling, is presented as a procedural model for Systems engineering. In fact, the RFLP approach is approximately equivalent to the descending side of the V model, which it includes. It highlights the fact of creating a functional model out of requirements engineering, whereby the main function is decomposed into sub-functions that can spread over many hierarchical levels.

The logical model describes interdependencies between the system components and therefore enables the right assembly of components to a final construct that can be simulated resp. verified. In short, the logical model connects the behavior models of the single sub-systems and system components, so they can interact with each other.

The physical model is represented by the representation (e.g. geometrical) of sub-systems and system components in the involved disciplines. Although the logical model is to be created before the physical modeling, the practice shows that behavior models can be generated from a physical model (e.g. geometrical model) in order to be attached to its corresponding component in the logical model. This approach can occur automatically when an integrated environment is used for SE (see Chap. 27). Otherwise, single behavior models can be create with own tools and assembled within a logical models, assuming that the relevant interfaces are implemented by the software that is used to design the logical model.

Biahmou et al. [42] have developed an approach for deriving behavior models from 3D models in MATLAB/SimMechanics that have been created with CATIA V5. After updating the geometrical model, the behavioral model is to be updated by the user. In order to free the user from this task and achieve a structured and right synchronization of changes in partial models of a system, the application of ontologies has been addressed to identify the update sequence of models [43].

Why has the RFLP-approach been necessary although the V model has been existing? Even though the V model has been provided for the development of mechatronics products involving mechanical, software and electrical/electronical engineering, it does not explicitly provide details on creating models as well as model boundaries in its initial form. However, an amelioration of Ott emphasizes systems thinking and traceability [44].

Furthermore, the V Model is compatible with well-known design methodologies such as the approach described by Pahl and Beitz, which recommend the identification of the main function and its fragmentation on many levels in less complex and manageable functions. Since the design is an iterative task, the many steps of the V Model or the whole V Model can be performed many times [11, 45].

The general processes of SE can be adopted by some business branches, companies or institutions. NASA has defined a SE process based on tasks integration and control as well as interfaces management, where both are to be performed during all the phases of implementation of a space mission. Other activities consisting of requirements engineering, system analysis, design and configuration definition and finally the verification, can be achieved sequentially [46].

At this stage, questions of importance are, what is needed except to the processes, to practice systems engineering? Which tools and methods are relevant, which organization structure and project management form are needed, how can SE be introduced in a company and which services should provide a platform for applying systems engineering?

9.3.1 Tools and Methods of Systems Engineering

The different disciplines that are involved in SE (e.g. mechanical engineering, software, Electronic/Electric) have different vocabularies and use different tools as mentioned above. Therefore, they need a common language as well as a common reference, that is, a common product model to tackle the aforementioned challenge of communication and coordination. The systems modelling language (SysML) is the most widespread solution for these issues. SysML is derived from unified modeling language (UML) and the relevant sub-set of UML (UML4SyML) has been enhanced in order to take the characteristics of general-purpose systems into account. SysML can be used for representing systems, which may include combinations of hardware, software, data, people, facilities, and natural objects [47, 48]. Models created with SysML can be exchanged using the XML metadata interchange (XMI) format, enabling also developers who use different tools to exchange model information [48]. Due to its organizational concepts of package, models and views, SyML enables the achievement of a paradigm shift in the modeling of complex systems of systems, from a document-based approach to a model-based approach.

The document-based SE Approach consists of keeping all relevant information in documents of different disciplines, depicting them in a hierarchical tree and defining how the system is to be used. One of the limitations of document-based SE

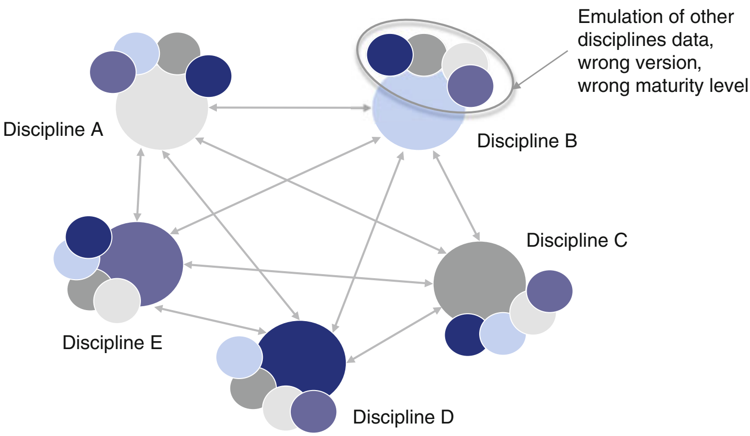


Fig. 9.5 Model creation in disciplines

is the difficulty to realize traceability and, therefore, to ensure the completeness of requirements. The link between requirements, design, engineering analysis and test information is often missing, leading, therefore, to a limited capability to manage change impacts.

Furthermore, document-based SE can lead to a significant lack of information across the disciplines involved in product development and to the use of inconsistent models, especially when each discipline is trying to create the missing model resp. information (see Fig. 9.5).

In contrary, model-based systems engineering (MBSE) is characterized by using a common model of the targeted system (see Fig. 9.6) to perform the tasks described on Fig. 27.3. That model may have references to distributed partial models. Alternatively, it can be associated with partial models that are used to represent the different artifacts (e.g. requirement model, functional model, logical model, geometrical model) and their relationships, in the same repository [49].

An important factor to be considered in the practice of MBSE is the representation of the variability, that for instance may be functional or physical. Since many details are not know or available in early development stages, it is important to agree on the right granularity of the common system model to support capturing of all relevant features as well as their relationships and therefore ensure traceability very early. Dumitrescu et al. [50] addressed this with a concept for introducing variability management on an intermediary level between vehicle features and component specifications.

Particular aspects of the global resp. common system model are called views or partial models, they can be generated from the common model by synchronizing the common parameters.

This means that a parameter model also is needed, in which implicit as well as explicit relationships of all involved partial models are represented. In the practice, a multidisciplinary team of well experienced specialists (e.g. designers) can define

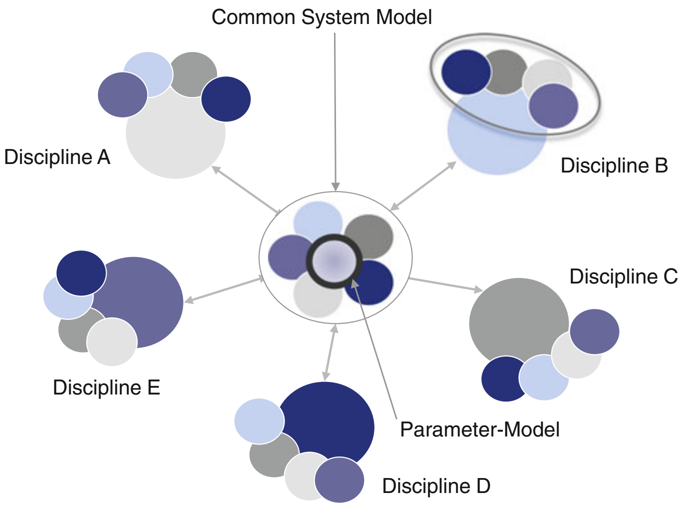


Fig. 9.6 Common system model

the parameter model. Even though a company is not applying systems engineering, each discipline generally use to work according to input-processing-output-principle (IPO). That is, a discipline receives data from another one, processes it to reach its own objectives, for instance to complete the geometrical design and at the end, provides their result—output—to other disciplines that are considered as data consumer in this case.

Further relevant methods of SE regard quality analysis and decision making support. Among others, conventional approaches such as Hazard Analysis and failure mode and effects analysis (FMEA), total quality management (TQM), Six Sigma, ISO/QS900x as well as software development quality systems based on capability maturity model integration (CMMI) or software process improvement and capability determination (SPICE), are used for quality management [51].

Equally, a multitude of approaches for technical decision making during early development phases are available [52]. However, the experience has shown that the partial models mostly are existing in proprietary formats with different modelling paradigms and therefore cannot be easily assembled to a global system model.

In order to apply SE without restrictions, standards are necessary for instance for model exchange or for enabling reuse or integration of components in third parties architectures. Therefore, the automotive open system architecture (AUTOSAR) specification has been elaborated for appliance by some companies that develop complex products (e.g. car makers, developers of engine control units, car software, development tools and microcontrollers) [53].

Hence the functional mock-up interface (FMI) has been developed as standardized interface to support model exchange between different tools for systems simulation, co-simulation, applications as well as the integration of models and their associated data into product lifecycle management (PLM). Software tools which have implemented the FMI standard can create an FMI-conform model library, called functional mock-up unit (FMU) (see Sect. 13.5) [54]. A further methodological approach for complexity management, especially for reducing the risk of failures of safety relevant components is the norm ISO 26262, which describes a procedure model to be applied [55].

Although more methods might be mentioned in such a chapter, for sake of brevity, only the SysML as well as interfaces are emphasized in this work. Reason for this orientation are the importance of the right modeling of the whole system in terms of representing the reality and impact of the communication in the project, as both represent success factors for SE. A valuable taxonomy of SE process standards as well as methods which describe how to achieve the tasks described in the SE process is given [48]. The taxonomy also includes architecture frameworks, systems modeling methods, modeling and simulation standards as well as interchange and meta modeling standards.

The tools necessary for SE have to support the realization of a major objective of product development, which is reducing cost through the reduction of physical Mock-ups. Visualization techniques and simulations of 3D models help for creating virtual mock-ups [56].

SE tools can be classified on two subjective levels: the early phase of product development corresponding to the descending branch of the V model, with the horizontal line of the V model corresponding to the detail design resp. discipline specific design. The second category of tools can be associated to the ascending phase of the V model. The core tools belonging to the first category are tools for creating a common system model, that integrates all partial models and their relationships (e.g. tools for SysML modeling). Hence tools are necessary for creating requirements models, elaborating functional models and common parameter models as well. A tool to model and simulate logical systems also is required.

Moreover, a platform or a tool has to ensure the traceability from the requirements over the specific design to the product recycling. That platform or tool should be equipped with analysis capabilities in order to provide quick answers to queries of a project leader. For instance, such an analysis tool should be able to show whether all requirements are fulfilled or not. Therefore, a project leader could check the impact of changes on requirements. Queries of project leaders could include risk analysis as well as cost calculation after design changes.

The discipline-specific tools that are used in the horizontal line of the V model already are part of today's tools landscape in the companies and can be considered as provided. However, the communication of these stand-alone tools is to be ensured though the use or implementation of appropriate connectors and enterprise architecture. The most aforementioned functions are available in tools that are on the market as stand-alone, but there are integrative platforms that propose environments that comprise many of the relevant functionalities. They are often equipped with interfaces that enable the integration of selected third party tools. The decision, whether an integrated environment or many stand-alone tools are to be connected depends on the company's SE strategy and the importance of the flexibility. Viewing from this perspective, SE can be applied without using an integrative development environment. A company can implement a specific architecture resp. environment with best-of-class tools of each category. For the ascending phase of the V model, manifold architectures, vehicle simulation and testing tools are commonly used for SE [26].

9.3.2 Organisation Structure and Project Management for SE

SE is applied by many companies, but the question is to know to which extent, as well as the impact of the existing organization structure on the practice of SE. The organization structures of carmakers have been component-oriented for a long time. Thus, the technical departments mostly spreads across the departments chassis, powertrain, interieur/exterieur, electric/electronics etc. [51]. Additionally to the component oriented structure, the experience has shown that a program resp. product oriented structure is implemented. Thus, a matrix organization is established in most cases. Project teams are made up of specialists from the different components departments (e.g. powertrain).

The success of SE project organized in that manner generally is strong dependent on the experience as well as the personality of the project leader. Success in this case is not limited to the question whether the result obtained by the project group is acceptable or not. The question is to know what would have been possible due to the skills of project members. Thus, the focal point is efficiency of the project team, adherence to delivery dates, product quality as well as stakeholder satisfaction. In order to take benefit of SE methods, it can be of advantage to move the location of the systems engineer according to Fig. 9.7. Doing so can help taking the focus away from the historical component-based development to the systems-based development.

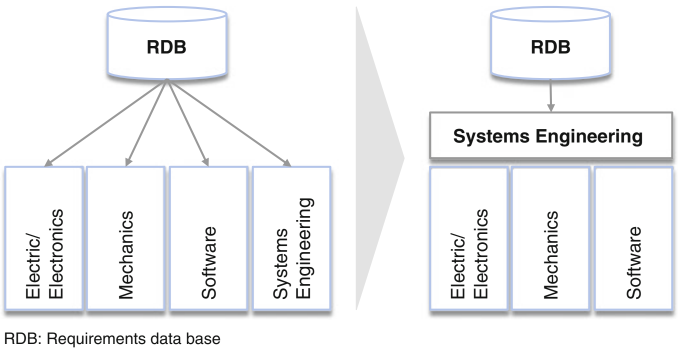


Fig. 9.7 Project organization for systems engineering

Applying SE requires therefore appropriate organization structures and corporate cultures. SE is not to be focused only on the product to be developed, instead the whole product creation landscape and processes including the manufacturing and the supply chain also are to be considered. This requirement is especially important when it is taken into account that many projects in aerospace or defense have been characterized by delays in delivering as well as increasing of initial unit costs. Sanders and Klein proposed an industrial V model that mirrors the conventional product V model in order to integrate manufacturing and supply chain considerations into the SE process [57].

It can be expected that similar approaches that consider further disciplines could improve collaboration and efficiency. For instance the development of Product-Service-Systems could be coupled to product SE on the same manner. Van Ruijven addresses the availability of a common framework that should include an ontology based on information models to support systems engineering. That framework should improve the communication within the project by defining SE processes based on the top level of an ontology that the author proposes. Additionally, information models of the breakdown of the process side of a system, the breakdown of the physical side of a system as well as of the work breakdown of a system are addressed to support that objective [58].

These proposals correspond partly to the need of a meta-model, that already has been mentioned by Winzer [4], to bring all involved parties to understand themselves without any ambiguities and emphasizes the importance of MBSE.

Moreover, merely the clear elaboration of SE processes surely will not be sufficient. The project management will have to emphasize the systems thinking at all stages of the project and should be supported by business services that orchestrate workflows in background and that process data from the relevant disciplines in order to provide up-to-date, integrative view data to the project management, but also to the disciplines. Such an overview is of great importance for project management in order to support decision making.

From a general point of view, Boehm et al. [59] have identified four key principles that are necessary to apply SE successfully: Stakeholder Value-Based System Definition and Evolution, Incremental Commitment and Accountability, Concurrent Multidiscipline System Definition and Development, and Evidence-Based and Risk-based Decision making. In order to emphasize the suitability of these principles, they are compared with lean SE principles as well as Hitchins' principles for successful SE.

As aforementioned, one could wonder why spectacular failures leading often to recall campaigns [14] are still been made in projects that have been driven by SE. One of the causes has been identified by Boy and McGovern as the lack of a human-centered focus in SE, that instead is claimed to be technology-centered and finance-driven. Thus, they propose to operate changes on rigid SE processes in order to provide the engineers with more flexibility, by applying Human-Centered-Design principles to achieve a Human-Systems Integration [60]. This calls for socio-technical leadership as an important skill for SE project leaders.

The idea of improving organizational aspects also has been identified by Winner [51], who pointed out both the staff qualifying for effective and efficient SE as well as subsequent modification of the management culture as part of challenges for SE in the automotive and supply industry.

Apart from the organizational structure of a company that can impact the success of systems engineering, the tools landscape is of great importance.

9.3.3 Architecture for Systems Engineering

Enterprises that decide to apply SE principles have to tackle the challenge of elaborating the architecture to be implemented in order to perform the processes mentioned above. An interesting approach for a SE platform consists of a tool federation instead of integration, whereby necessary information is shared between loosely coupled models (see Sect. 9.4).

A similar approach has been addressed by Bartelt et al. who proposed a software architecture resp. middleware for the configuration of simulation scenarios. The simulation scenarios can be integrated in simulation modules. The simulators communicate through the SimBus, that is an architecture on the basis of a CORBA-like platform. SimBus connectors allow the communication of applications [61]. The solution is specialized on simulation purposes, however SE addresses a broader spectrum.

A system framework for conducting SE is based on a complex architecture, since it is in turn a system of systems. Therefore, elaborating them requires the consideration of: Autonomy, Complexity, Diversity, Integration strategy, Data architecture and System protection [62]. Among other critical factors, robustness and alignment to business processes and technology are to be taken into account.

Explaining all processes and principles of architecture design would go beyond the scope of this chapter. However, apart from ISO/IEC 42010 (2011) [63] that provides standards for architecture description for systems and software, there are well-known commercial, defense and government frameworks [64] that can be used as guidance when elaborating enterprise architecture:

- the Zachman Framework for Enterprise Architecture [65],
- the Department of Defense Architecture Framework [65],
- the Federal Enterprise Architecture Framework [65],
- the Treasury Enterprise Architecture Framework [65],
- the Open Group Architectural Framework [65], and
- the NATO architecture Framework [66].
- An assessment of the five first frameworks by Urbaczewski and Mrdalj [65].

9.3.4 Summary Evaluation

Even though SysML at this moment is the language that is established to create common understanding models for SE, this term is still ambiguous among experts (e.g. SE from academia and industry). This calls for the necessity of harmonization for instance regarding basic terms such as function and behavior [67, 68].

Many companies still take the risk of considering the different phases of complex products creation as sequential tasks. They assume that the periodical information exchange between the disciplines is sufficient. This can lead in most cases to inconsistencies. Instead, a concurrent engineering approach is to be followed, ensuring a traceability between all the models and phases involved [69].

Moreover, many disciplines still provide other collaborating disciplines with data without an explicit and documented procedure. Preparing and providing data to others is generally a task on top of the daily job that consists of performing a progress in the original discipline, e.g. the design. Thus, requests from another discipline for receiving data can take a long time before being processed. It can cause the tendency to use obsolete data or to try creating data for own analyses.

9.4 Concept for Functional Blocks to Support SE

To bridge the disadvantages mentioned above, an alternative concept can be implemented, in which data and services sharing as well as the domain authority are highlighted. Thus, each discipline involved in product creation can access data of another discipline through sharing mechanisms, whereby re-creation of data is useless. A proposal of functional features to support this way of working is presented for companies as guidance for enabling Systems Engineering.

9.4.1 Selected Requirements

The three most important problems originating from collaborative work have been identified by Königs et al. [70] as follows: inconsistent, hard-to-retrieve or outdated data across the engineering departments, low transparency about changes and decisions due to non-existing or non-available information and low transparency about the impacts of changes due to missing documentation of dependencies. This traceability issue also has been addressed by many works [21].

Further requirements for a successful SE architecture are shown in Fig. 9.8 from an industrial study [1]. It reveals among others, that the extended enterprise as well as customers are to be taken into account and that the visualization of changes as well as using common data format are needed [71]. Thus, innovative techniques that include knowledge management can be applied to optimize the collaboration with the extended enterprise [72].

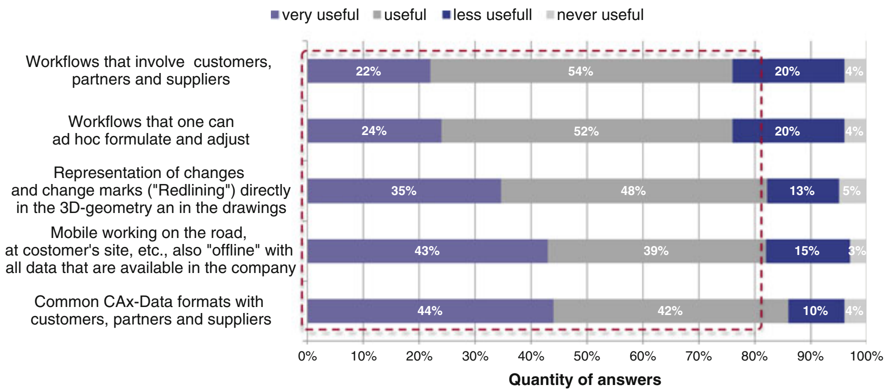


Fig. 9.8 Requirements for a successful development [1]

In fact, suitable interfaces are necessary for software communication, since the disciplines involved use different applications. Besides, neutral data formats such as STEP AP 214 (ISO 10303) are required due to the fact that the different software components involved often create proprietary file formats. A neutral file format can ease information exchange between many applications, because it helps avoiding bidirectional conversions.

Data exchange however, is not sufficient for an effective Systems Engineering. One of the challenges consists of applying data sharing; a use case could be a control system engineer generating a geometrical model from the global product in the suitable level of maturity in order to run a distributed simulation, in which both the geometrical modeler and the control system modeller are communicating.

Moreover, a SE platform should implement the most widespread software interfaces that are necessary to integrate further, third party tools.

9.4.2 Concept

A real prerequisite for SE is the alignment of methods and models of the disciplines involved in product development. Investigations that are performed by an arbitrary discipline do not need only a data set of another discipline as input, instead additional meta-data such as the maturity level are necessary.

Thus, a basic methodological work to be performed consists in gathering the relations between the disciplines models and to create additional stage gates that will be used to characterized data needed by other disciplines. Doing so enables the definition of fix or ad hoc workflows, since the stage gates can be used as attribute to identified the data to be retrieved. Furthermore, consistency also can be assured by synchronizing partial models of corresponding maturity levels.

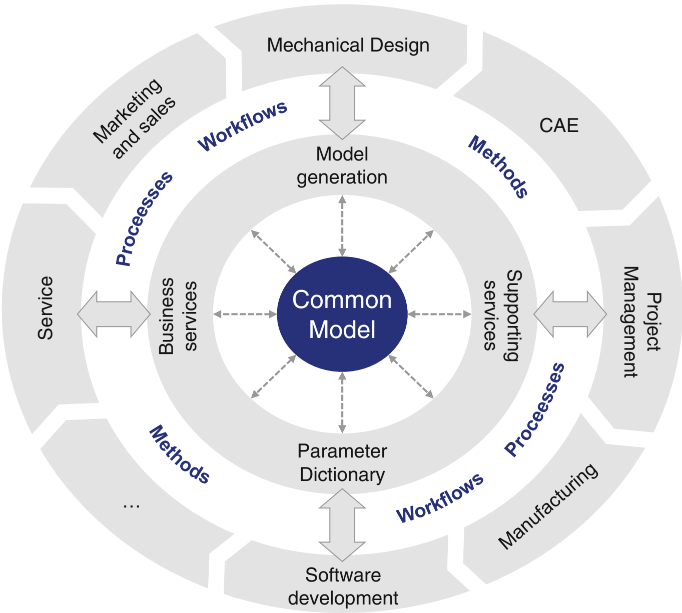


Fig. 9.9 Types of services needed for systems engineering

In order to fulfil the requirements mentioned above, a service-oriented architecture can be implemented with the functionality presented on Fig. 9.9. Thus, the disciplines can apply collaborative processes and methods in order to optimize the way of working according to the global product and not to a single aspect.

Model generation services can be used to generate an up-to-date, contextual partial model. They can be crucial when it comes to configure models for studies, especially when design of experiments are to be realized. A very simple example consists of a suitable geometrical model being automatically generated from the common system model in order to perform a finite element analysis (FEA), whereby the relevant specific attributes such as the level of maturity or the release status would have been taken into account.

Since an SE platform is supposed to be opened to external enterprises, encapsulating of the company's know how is an important criteria. Business services are to be identified, supported, and aligned with the products to be created as well as the processes. Thus, the services shall be defined on the basis of formulated use cases. The business services shall provide specific functions that are based on a service-oriented architecture, whereby many systems that are transparent for the user can be triggered in the background.

An example of business services is realized by lifecycle management services that shall provide all the generic functions of a product data management system (PDM) such as enabling configuration management, versioning and change management. Lifecycle management services shall ensure not only the update of the common model in the correct sequence, but also its integrity. Thus, modified partial models shall be synchronized with the common model only when the modifications have been performed by specialists.

However, the non-specialists who are using and might modify an arbitrary partial model for studies shall be enabled to store and share the results of these studies as well as to record their decisions and rationales. To ensure the quality of studies results, each discipline will have to define a range of admitted models as well as attributes modifications, which a data consumer is allowed to perform.

Additional business services may be among others workflow services, right management services, intellectual property protection services and assessment services. The latter can help to examine important criteria of the product to be created, for instance cost-effectiveness or even fuel consumption of a car. Assessment services can, therefore, support the project manager in decision making.

A parameter dictionary is also required to represent the relationships between the different partial models as well as their explanations. Such a dictionary should include a parameter repository based on the parameter model described in Sect. 9.3.1. Depending on the implementation, the parameter model can be indispensable for updating the different partial models after a change has occurred in one model. The explanations provided by such a dictionary can help improving the understanding of terms in the different disciplines involved. For this purpose, the team in charge of defining the work methodology shall provide an ontology as a formal representation of a set of concepts to create a common meta model for the relevant domain, or to adopt an existing meta model [58]. The meta-model shall help all disciplines involved to have a common language and a common reference.

9.5 Introduction of SE in a Company

The introduction of SE in a company can be organized in many phases depending among others on the company's internal culture and organization. Figure 9.10 depicts a recommendation based on own experiences.

In the preparation phase, the analysis consists in making a self-assessment in order to document what are the strengths and the weak points of the actual way of working. The real work processes are to be considered instead of an eventual process description that might not be applied in daily work.

Based on the results of the process analysis, a SE vision resp. target can be formulated. Assuming that the SE vision is to be realized, the SE analysis phase can start. In this phase, SE preliminary processes are to be performed. This consists in organizational tasks, e.g. installing a project steering committee and providing a broader circle of relevant employees with general information based on preliminary analysis results and the consequential need for introducing SE.

The steering committee is mainly in charge of defining a SE strategy to reach the vision defined in the preparation phase. The strategy has to provide a clear path to be followed, e.g. defining how the extended enterprise is to be taken into account,

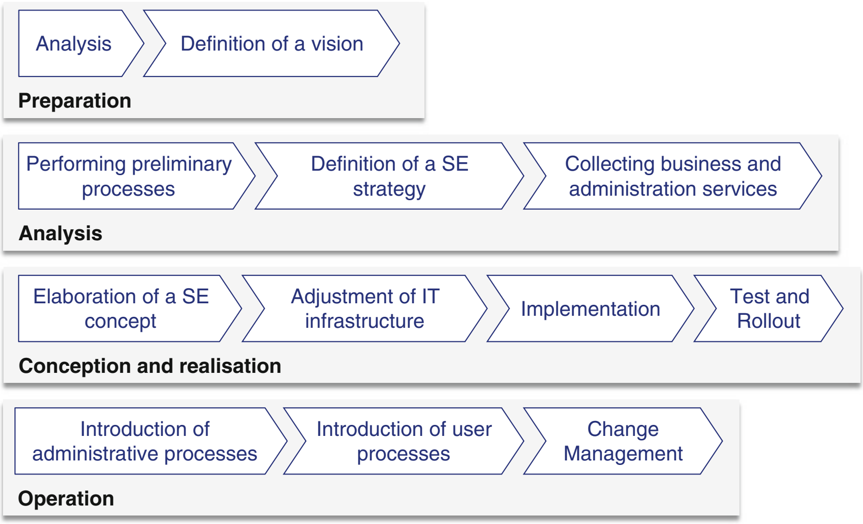


Fig. 9.10 Recommended phases for SE introduction

whether internal workflows and services are to be made available for extended enterprise. Defining the eventual SE levels to be realized is a strategic decision; SE could be realized first to an intermediary level with less functionality. Afterwards, a second or later level with more or full functionality could be implemented based on the experience gained at first levels. Strategic decisions in this context also consist of defining e.g. which types of systems to be used: opened systems, commercial-off-the-shelf-tools (COTS), integrated environments that provide the most functionality needed or taking best-of-class tools and connecting them. Outsourcing strategy and make-or-buy-strategy also shall be considered.

A further task that could be assigned to the analysis phase is the collection of use cases from the specific disciplines involved in the product lifecycle and complementing them with administration use cases.

During the concept and realization phase, requirements are derived from the collected use-cases and business as well as administration services are formulated. A functional architecture of the SE environment is to be defined, in which supporting services are considered. The methods of the different disciplines involved are to be aligned in order to take a higher benefit of SE. Thus, the concept considers not only the necessary software, but also the methodological way of working.

Depending on the SE architecture that has been defined, adjustment of the IT architecture might be necessary, e.g. additional leased lines could be necessary to guarantee some business services, especially when it comes to providing the extended enterprise with internal workflows. Further tasks after the concept and realization phase are the SE implementation and the test and rollout of the solution.

In the operation phase of the system administrative processes are introduced, system settings in tools can be adjusted and necessary accounts can be created in order to enable user processes. Latter consist in using the business services that are implemented in order to perform specialist tasks. The change management is a continuous process that takes into account the fact that the SE environment can be improved based on change requests that could be introduced by stakeholders.

9.6 Case Studies

The practical value of SE is illustrated with the following two use cases.

9.6.1 SE in Aerospace—Investigating Force Fighting on an Aileron

Vuillemin et al. describe an application of model based SE for a force fighting issue in the aerospace domain. The challenge to be tackled consists of investigating the system behaviour when two forces acting in the same direction or in opposite directions are applied on a surface. This can be caused among others by errors such as signal conversion errors or by adjustment tolerances [73]. Figure 9.11 depicts a schematic presentation of the system. In order to solve the issue aforementioned, the RFLP approach has been applied.

Thus, the systems requirements have been formulated by the authors in textual form. The next step has been the functional analysis. The mission of the system as well as the functions and corresponding I/O interfaces have been determined and the connections of functions as well as their sequences have been modeled [73].

A logical architecture of the aileron has been created as a block diagram on which all relevant components of the system are represented. The components of

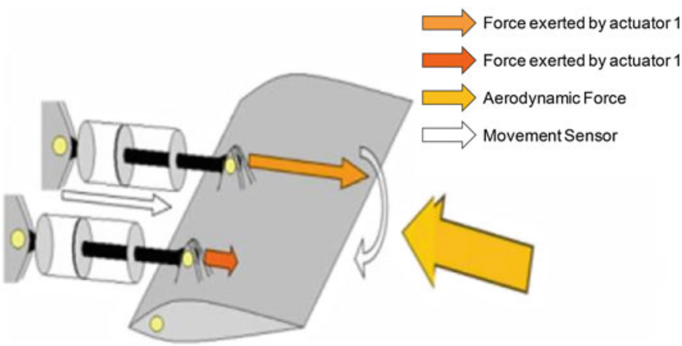


Fig. 9.11 Two forces acting on a surface [73]

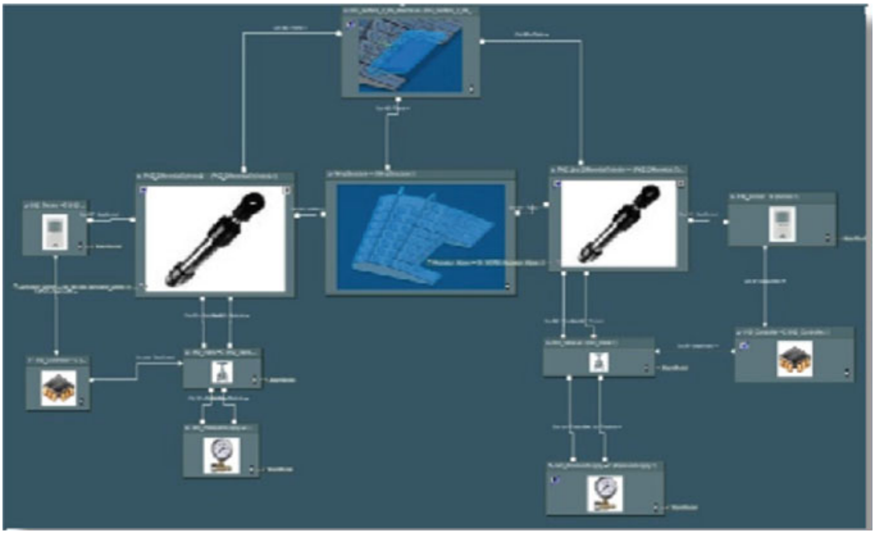


Fig. 9.12 Logical model of the aileron system [73]

the logical architecture (e.g. physical model, control system) are to be attached to behavioral models in order to enable a whole system simulation.

Since the global simulation is using a Modelica platform in background, the control component that has been created in MATLAB/Simulink is used for generating a functional mockup unit (FMU), as neutral format. Latter is attached to the logical model (Fig. 9.12) [73].

In order to consider additional models such as FEA models in the simulation, they have been simplified, then represented in the SID format, that can be imported by Modellica libraries [73]. In fact, SID files can be generated from the most commercial FEA programms with order reduction methods and animation information which is used by the Flexible Bodies library of Modelica. A multi-physic simulation of the whole aileron can be started, in which the deformations of the aileron are visualized. When movements of the aileron are required, the actuators receive hydraulic power from the hydraulic components they are connected with and therefore, the aileron is moving according to the impact of both fighting forces.

Several parameters that are important for the system, for instance the pressure, the position and the angle are represented in plots and therefore can be assessed by engineers. The system validation is performed through checking the links existing between the formulated requirements and each of the models created, that are the functional, the logical and the physical models. The objective is to determine potential discrepancies.

9.6.2 Hubble Space Telescope Systems Engineering

The Hubble Space Telescope (HST) is an observing system of systems that produces imaging, spectrographic, astrometric and photometric data. The HST has been developed based on SE principles, therefore its facet as a valuable result of SE has been highlighted in a case study originally provided by Mattice [74].

The HST case study has been guided by the Friedman-Sage Framework that enables the presentation of contractor, shared and government responsibilities for nine concept domains. These domains are spanning from the requirement definition and management over the systems architecting and conceptual design up to system and program management [74, 75]. For sake of brevity, this structure is not followed in this document.

Developing the HST has been necessary because the composition of the atmosphere limits the resolution of telescopes that are located on earth. Therefore, an alternative system was necessary, that should be located in space, outside the atmosphere.

The HST has been deployed 1990 in low-Earth orbit (600 km). It has been designed to observe the space permanently and to be maintainable, therefore adjustments can be performed during regular servicing missions. For this purpose, it has been equipped with necessary components such as grapples and handholds for control [74], but also with diverse components that enable the communication as well as an external control. Figure 9.13 depicts the major components of HST.

9.6.2.1 Requirement Definition and System Specification

The main requirement to the HST has been to provide relevant data to astronomers in order to help them conducting research in their scientific domain. Astronomer-scientists who took the role of a customer defined the requirements regarding the capability of HST. Among others, they described the observations that the HST should enable, when and by whom observing operations were to be performed [74]. Requirements also regarded the external controlling of the HST, design, development, in-orbit operations, maintenance and so on.

Due to conflicts between the astronomer scientists and the NASA in this phase, the HST-linked institute was created as neutral instance for managing HST project. It was in charge of defining the location, the research agenda, the scientific instrument requirements as well as playing a key role in HST ground and space operations [74]. An important lesson learned from this phase is the fact that the customer or user should be involved from the beginning throughout the project in order to get success.

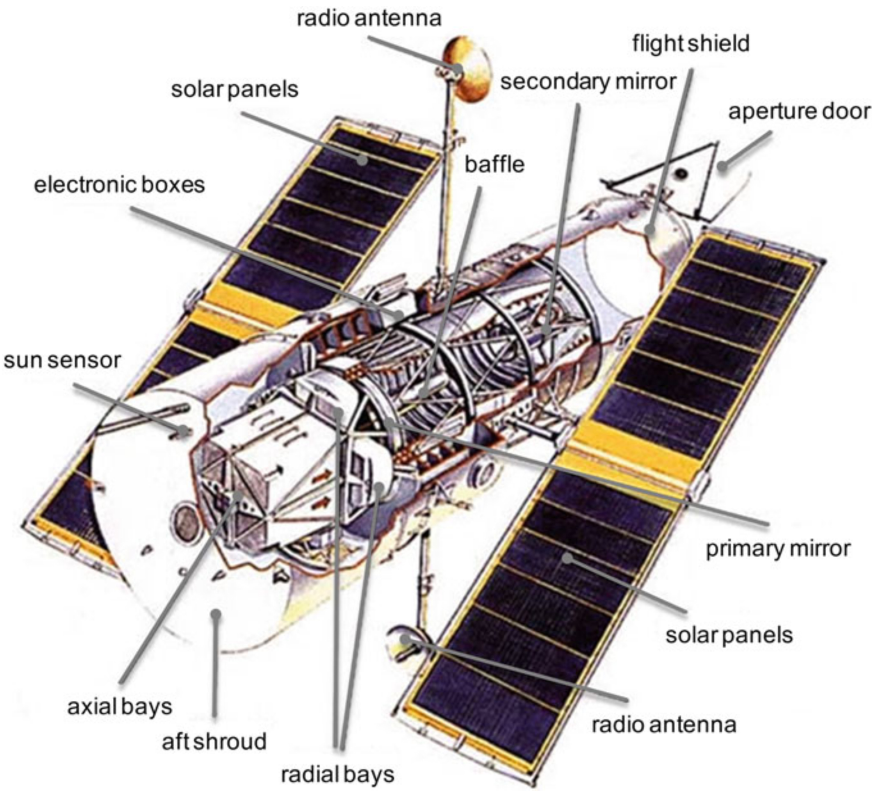


Fig. 9.13 Main components of HST [74]

9.6.2.2 Systems Architecture and Conceptual Design

Pre-program trade studies that were provided by contractors and NASA centers offered important inputs for HST project, since the submitted technical concepts addressed critical requirements and feasibility issues.

However, it is important to find the right balance between the different requirements such as costs and technical functions. For instance, some trade studies claimed the presumed advantages of reducing the HST mirror from 3 to 2.4 m in order to reduce costs. Doing so was assumed to positively impact manufacturing, test and assembly whereby the transport of the HST in the space still would not be impeded. The reduced size also should be suitable for fulfilling the requirements regarding light gathering, optical accuracy, pointing as well as stability control [74]. Analysis however had established some drawbacks related to that proposal, such as the reduction of the light collecting capability to the third, the reduction of the resolution capability and even more, although a weight reduction would have been achieved.

Furthermore, additional concepts including operation for launch, deployment and servicing was elaborated with cost trade-offs. The conceptual design was refined with costs and requirements for detailed design, development and construction were identified. Finally, the implementation of the system architecture has been more impacted by technical requirements rather than the cost projections [74]. This phase was characterized among others by the following activities:

- Risk, cost, schedule and configuration management
- Independent review and payload specification groups
- Case-dependent simulation, laboratory and ground testing prior to initial flight and on-orbit repair
- Definition of relative roles and contributions of involved stake holders.

9.6.2.3 Design

The high requirements (e.g. tolerance requirements) to the mirror that contains the primary mirror (Fig. 9.13) have been important factors to be considered in this phase. Innovative solution approaches were introduced in order to counter the weight of the mirror and therefore achieve the goal of zero gravity of the mirror for testing purposes [74].

Further critical points have been among others, the engineering and assembly of main sub-systems and components as well as guidance sensors.

9.6.2.4 System Integration

Reducing the mirror size from 3 m to 2.4 revealed itself to be a substantial modification, since other main components were impacted. The envelope of the optical components of the HST had to be redesigned to meet the new dimensions. This led to many component pairs not anymore linkable with each other. This challenge has been tackled using kinematic joints to dynamically isolate the components [74].

In order to perform the system integration of the HST physical, structural, electrical, optical, electronic, thermal control, power as well as operational software/hardware domains were involved. The phases of pre-ascension, ascension and post-ascension were distinguished [74].

Furthermore, instrument (e.g. mirror) specific requirements, e.g. for packaging, power, thermal control and orientation as well as additional functional requirements were to be taken into account and considered from the instrument level down to the component level.

A high amount of functions were simultaneously monitored and were able to be controlled not only on Earth but also from the shuttle that would take the HST on orbit, after it has been deployed [74].

Another important issue regarding system integration has been the weight allocation and management. The weight of subsystems was to be tracked in order to organise launching as well as maintenance missions [74].

In order to fulfil the weight requirements, an important effort (e.g. creation of weight reference plans, documentation) has been necessary to preserve flexibility from the development, fabrication, integration, deployment up to the maintenance phases [74].

The system integration included more than the weight dimension and had required a high degree of discipline, documentation and communication involving not only humans, but also machines. The importance of system integration is addressed by Langford [38] in several facet, whereby seven integrations principles are elaborated as recommendation: alignment, partitioning, induction, limitation, forethought, planning and loss. It may vary depending on the type of acquisition, the amount of detail in the system design, and the degree of specificity for the concept of operations.

9.6.2.5 Validation and Verification

Many options have been discussed for HST validation and verification. For instance, the alternatives of performing verification on ground or in the real operation conditions, that is, in the space, whereby costs and risks were to be considered. Performing also incremental tests were compared with conducting all-up tests that the NASA already had successfully experienced in former projects. The higher risks associated with HST led to thinking about conducting a complete system vacuum thermal test in a chamber, thus providing a realistic test environment [74].

Although this type of test only could generate new issues in term of costs, it was decided to adopt them in order to save long term costs since a system that is not tested enough might have generate more maintenance efforts and therefore more costs.

An up-and-running test program was performed during 30 days and showed the weaks of the system, such as the unreliability of power supply. Due to the tests results that were not satisfying, it also was discussed whether it would have been better to conduct tests cycles, in which incremental tests of the design would help reaching a final status that would satisfy the requirements [74].

These discussions highlight the dilemma in which the project members have been in, but also the key role of an early system integration and validation, which are supported by a consistent application of SE.

9.7 Discussion

SE can be apprehended as a means for mastering complexity while developing systems of systems. It defines processes as well as methods to be performed, that might suggest a certain rigidity. However, SE grants enough freedom in the detailed realization of the high level processes. For instance, parallelization can be performed during requirement management, functional model development and so on. Besides, templates can and should be used in the different stages of SE in order to enable the parallelization of tasks across the activities described by the SE process. For instance, a logical system model can be elaborated using templates and therefore would be available for early studies, before the functional model has been detailed.

Furthermore, SE specifies that the inputs for the creation of physical models are to be provided by the logical model that represents the systems view (Fig. 9.7). This enables therefore the parallel creation of domain specific models instead of a sequential development of the physical models, that would be driven by the mechanical design. Since concurrent engineering addresses the parallelization of diverse phases and processes of product development, it can be considered as a complementary for systems engineering.

A further aspect that deserves to be discussed is the procedure of enabling SE in a company. Although the process of SE as well as tools and methods have been presented and referenced, an important factor remains the economic considerations, since the leading managers often would ask about the return of investment (ROI) when it comes to introduce new methods and systems. SE surely would not make an exception. Therefore, the SE strategy should emphasize the needed SE capabilities and if necessary, implement SE following an incremental approach. Especially the elaboration of an SE architecture should be approached with this perception in mind. While some software vendors are proposing integrated environment, a dedicated SE architecture that takes legacy applications into account could be more advantageous from a financial and technical point of view.

Integrated environments certainly enable a quick start-up, however, they can be characterized with limitations. This often is the case when a data set to be processed is existing in a proprietary format of competitors. Therefore, the support of standards is crucial (e.g., FMI, Spice, CMMI, ISO61508, ISO26262, Spice, DO178C, DO254, FDA, GAMP etc.). Even though some initiatives are working for boosting the openness of systems (e.g. Code of PLM Openness Initiative [76]), there is still a lot to do in order to integrate different systems.

Furthermore, while most scientists are agreeing in the meantime that MBSE is one of the core elements of SE, questions still remain about its formal creation and the language to described it. SysML certainly is widespread, however, investigations have shown that all the engineers involved in the product development are not experienced with it. Besides, further languages are existing. This calls for a stronger integration of SE in academia, not only for special programs, but also for all technical studies.

Considering the background of zero error products, manufacturing as well as other disciplines related to product creation, should be handled with the same engagement as product SE, since even an excellent design does not exclude making failures during manufacturing. This has been observed for instance in the HST case study, that identified the aberation failure [74] of HST after 1990 launch as a failure arising from polishing operations during manufacturing. Product-Service Engineering also should be considered from the first product idea onward.

Looking somewhat further ahead, the complexity will increase further and the question whether a general or a specific SE-approach should be used will remain. Next generations of systems of systems are planned to be self-optimizing, adaptive and even autonomous. Many carmakers as well as well-known software companies already have presented early prototypes of autonomous cars. This will bring challenges to be tackled by SE. Furthermore, the advances that might be expected in the development of electric vehicles, such as the communication of cars with power supply infrastructure as well as with diverse further systems (e.g. other cars) and the necessity to protect such systems of systems against hackers are some examples of complex challenges for SE. Sustainability calls for new materials and concepts, more and appropriate simulations will be necessary.

Not only the systems to be created using SE are to be considered, but also the enterprise processes for performing SE. In this case, the tends for bring-your-own-device, the integration of social media in the product development, whereby customers experience and requirements are captured, surely necessitates appropriate SE approaches. It can be expected that customer will not influence only the product development, but also its manufacturing, in regards of rapid prototyping technologies being more and more available for individuals [71, 77–79]. Thus, a manufacture-it-yourself mentality could call for new ways of developing products.

All the factors mentioned above as well as the expected influence of customers require efficient decision making tools and processes. The aforementioned challenge of human-centered design, the standardization of the vocabulary in SE as well as organizational changes in companies to support SE also are to influence the way of working [80]. In order to conform to SE principles such as the elaboration of a functional model, going back from existing physical systems to their functional models is supposed to be necessary, since these models are to be integrated in the functional model of their advisory system of systems [81].

The trend to shorten product lifecycles as well as personal proposals or recommendation to customers due to increasing capture of customer behaviour, for instance with cookies or game consoles, creates more challenges for systems integration, knowledge management and variability management [82]. Since companies are working more and more in cooperation (e.g., cooperation for developing car batteries or composite materials) and using clouds services, regarding the growing tend for mobile offices, intellectual property protection, especially enterprise rights management will influence today's conventional way of working.

9.8 Conclusions and Outlook

Product complexity is multidimensional and consists of product as well as process complexity.

Regarding product complexity, the number of functions as well as components has increased in the last decades. New functions have been created and assigned to components and new components (e.g., electronic components of a passenger car) have been created, for instance, for safety purposes. Therefore, additional interfaces are to be considered on the one hand between the sub-systems of the car and, on the other hand, between the car and its environment, which includes passengers, especially when thinking of topics such as smart car, connectivity, and so on.

Product complexity is accompanied by process complexity, which is characterized by the use of more and more complex tools, interfaces and specifications when thinking in term of compliance [83]. In some cases, process discontinuities have led to important delays of the start of production (SOP) of well-known airplanes and cars models.

Process complexity is not limited only to development processes, since industrial processes also are impacted by the requirements that are to be fulfilled by products as well as company internal objectives such as target zero defect, lean, green and compliant manufacturing [84]. Besides, innovation leads, for instance, to applying new processes and materials, which may imply a higher degree of complexity like in smart factories.

In order to manage complexity while developing complex industrial products such as cars and planes, which of course are systems of systems, the techniques of SE can be applied. They are appropriate means that help the companies either to cope with complexity, to manage it or to reduce and eliminate it [79].

SE relies on models and methods that are to be elaborated for solving a problem and to reach a target of the system, which is ultimately satisfaction of stakeholders [85]. Verification and validation are necessary to ensure that requirements have been fulfilled in a manner that satisfies the stakeholders [59].

Since many disciplines are involved in the creation of complex systems, determining a meta-model of the system-of-interest as well as sharing knowledge are essential. Clear and defined interfaces between disciplines are necessary in order to enable information sharing. That common understanding is fundamental for systems thinking.

This chapter is an attempt to provide a deep understanding of Systems Engineering. The SE process as well as relevant tools and methods have been presented. Besides, proposals have been described for implementing a SE platform and for introducing SE in a company.

References

1. Müller P, Pasch F, Drewinski R, Hayka H (2013) Kollaborative Produktentwicklung und digitale Werkzeuge: Defizite heute – Potenziale morgen. Fraunhofer-Institut Produktionsanlagen und Konstruktionstechnik IPK
2. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manage 6(4):307–323
3. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Syst Manage 6(1):7–24
4. Winzer P (2013) Generic systems engineering. Springer, Berlin

5. Lindemann U, Maurer M, Braun T (2009) Structural complexity management. An approach for the field of product design. Springer, Berlin
6. Schuh G, Schwenk U (2001) Produktkomplexität managen. Hanser, München
7. Hitchins DK (2003) Advanced systems thinking, engineering and management. Artech House, Boston
8. Tolk A, Adams KM, Keating CB (2011) Towards intelligence-based systems engineering and system of systems engineering. In: Tolk A, Jain LA (eds) Intelligence-based systems engineering. Springer, Berlin
9. Frezzini FR, Sachan R, Azimi M (2011) Review of systems engineering scope and processes. In: Kamrani AK, Azimi M (eds) Systems engineering tools and methods. CRC Press, Boca Raton
10. Lamb CMT (2009) Collaborative systems thinking. An exploration of the mechanisms enabling systems thinking. PhD thesis, Massachusetts Institute of Technology, Cambridge
11. Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung. Springer, Berlin
12. Lindemann U (2005) Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerechtanwenden. Springer, Berlin
13. Ehrlenspiel K (2003) Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit. Hanser, München
14. NN (2014) Chrysler ruft rund 870.000 Geländewagen zurück. Spiegel-Online. http://www.spiegel.de/auto/aktuell/jeep-cherokee-chrysler-ruft-rund-870-000-gelaendewagen-zuruecka-962246.html. Accessed 2 Apr 2014
15. Haskins C (2011) International council on systems engineering (INCOSE) systems engineering handbook. A guide for system life cycle processes and activities. V. 3.2.1, INCOSE-TP-2003-002-03.2.1
16. Hitchins D (2007) Systems engineering: a 21st century systems methodology. Wiley, Chichester
17. Moser AH (2014) Understanding complex systems. A case study in space industry. Springer International Publishing, Switzerland
18. Szabo C, Diallo SY (2011) Defining and validating semantic machine to machine interoperability. In: Tolk A, Jain LA (eds) Intelligence-based systems engineering. Springer, Berlin
19. Fumarola M, Seck M, Verbraeck D (2011) A simulation-based systems design in multi-actor environments. In: Tolk A, Jain LA (eds) Intelligence-based systems engineering. Springer, Berlin
20. O'Shea J, Zuhair B, Keeley C (2011) Systems engineering and conversational agents. In: Tolk A, Jain LA (eds) Intelligence-based systems engineering. Springer, Berlin
21. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manage 7(3/4):324–347
22. Albus J et al (2008) Intelligent control of mobility systems. In: Prokhorov D (ed) Computational intelligence in automotive applications. Springer, Berlin
23. Prokhorov D (2008) Neural networks in automotive applications. In: Prokhorov D (ed) Computational intelligence in automotive applications. Springer, Berlin
24. Gandhi T, Manubhai Trivedi M (2008) Computer vision and machine learning for enhancing pedestrian safety. In: Prokhorov D (ed) Computational intelligence in automotive applications. Springer, Berlin
25. Bergasa LM, Nuevo J, Sotelo MA, Barea R, Lopez E (2008) Visual monitoring of driver inattention. In: Prokhorov D (ed) Computational intelligence in automotive applications. Springer, Berlin
26. Kamrani AK (2011) Genetic-algorithm-based solution for combinatorial optimization problems. In: Kamrani AK, Azimi M (eds) Systems engineering tools and methods. CRC Press, Boca Raton
27. Garlan CM, Colombi J (2011) Systems engineering case studies. In: Kamrani AK, Azimi M (eds) Systems engineering tools and methods. CRC Press, Boca Raton

28. Cavalieri S, Pezzotta G (2012) Product-service systems engineering: state of the art and research challenges. Comput Ind 63:278–288
29. Peruzzini M, Germani M (2014) Design for sustainability of product-service systems. Int J Agile Syst Manage 7(3/4):206–219
30. Kossiakof A, Sweet WN, Seymour SJ, Biemer SM (2011) Systems engineering principles and practice, 2nd edn. Willey, Hoboken
31. Dikerson CE, Mavris DN (2008) Architecture and principles of systems engineering. CRC Press, Boca Raton
32. Pineda RL, Smith ED (2011) Functional analysis and architecture. In: Kamrani AK, Azimi M (eds) Systems engineering tools and methods. CRC Press, Boca Raton
33. Ryschkewitsch M, Schaible D, Larson W (2009) The art and science of systems engineering. Syst Res Forum 03(02):81–100
34. Rupp C (2009) Requirementsengineering und –management; Professionelle, Iterative Anforderungsanalyse für die Praxis, 5th edn. Carl Hanser Verlag, München Wien
35. Gilb T (2005) Competitive engineering—a handbook for systems engineering requirements engineering, and software engineering using planguage. Elsevier, Oxford
36. Buede DM (1999) Functional analysis. In: Sage AP, Rouse WB (eds) Handbook of systems engineering and management. Wiley Inc, New York
37. Palmer JD (1999) Systems integration. In: Sage AP, Rouse WB (eds) Handbook of systems engineering and management. Wiley Inc, New York
38. Langford GO (2012) Engineering systems integration theory, metrics, and methods. CRC Press, Boca Raton
39. Shabi J, Reich Y (2012) Developing an analytical model for planning systems verification, validation and testing processes. Adv Eng Inform 26(2):429–438
40. Kolonay RM (2014) A physics-based distributed collaborative design process for military aerospace vehicle development and technology assessment. Int J Agile Syst Manage 7(3/4): 242–260
41. Sop Njindam T, Platen E, Paetzold K (2012) Modellbasiertes systems engineering Zur Frühzeitigen Absicherung Komplexer Multidisziplinärer System. Tag des Syst Eng, pp 271–282
42. Biahmou A, Fröhlich A, Stjepandić J (2010) Improving interoperability in mechatronic product development. In: Thoben KD et al (eds) Collaborative value creation throughout the whole lifecycle. Proceedings of PLM10 international conference, Inderscience, Geneve
43. Kuhn O, Liese H, Stjepandić J (2011) Methodology for knowledge-based engineering template update. In: Cavallucci D, Guio R, Cascini G (eds) Building innovation pipelines through computer-aided innovation. Springer, Berlin, pp 178–191
44. Ott S (2009) Konzept zur methodischen Systemmodellierung in der anforderungsgerechten Produktentwicklung. PhD thesis, Universität Wuppertal
45. Maurer M (2013) Automotive systems engineering: a personal perspective. In: Maurer M, Winner H (eds) Automotive systems engineering. Springer, Berlin
46. Aguirre MA (2013) Introduction to space systems: design and synthesis. Springer Science +Business Media, New York
47. Weilkiens T (2008) Systems engineering with SysML/UML: modeling, analysis, design, 2nd edn. Dpunkt Verlag, Heidelberg
48. Friedenthal S, Moore A, Steiner R (2012) A practical guide to SysML: the systems modeling language, 2nd edn. Morgan Kaufmann, Waltham
49. Brown B (2011) Model-based systems engineering: revolution or evolution? Thought Leadership White Paper, IBM Rational
50. Dumitrescu C, Tessier P, Salinesi C, Gerard S, Dauron A, Mazo R (2014) Capturing variability in model based systems engineering. In: Aiguier M, Bretaudeau F, Krob D (eds) Complex systems design and management. Springer, Berlin, pp 125–139
51. Winner H (2013) Challenges of automotive systems engineering for industry and academia. In: Maurer M, Winner H (eds) Automotive systems engineering. Springer, Berlin

52. Levandowski C, Raudberget D, Johannesson H (2014) Set-based concurrent engineering for early phases in platform development. In: Cha J et al (eds) Proceedings of 21th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 521–530
53. Kindel O, Friedrich M (2009) Softwareentwicklung mit AUTOSAR. Grundlagen, Engineering, Management für die Praxis. dpunktVerlag, Heidelberg
54. NN (2013) Functional mockup interface (FMI)—version 1.0. https://www.fmi-standard.org/downloads. Accessed 15 May 2014
55. NN (2013) ISO26262, ISO. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54591
56. Dineva E, Bachmann A, Moerland E, Nagel B, Gollnick V (2014) New methodology to explore the role of visualisation in aircraft design tasks: an empirical study. Int J Agile Syst Manage 7(3/4):220–241
57. Sanders A (2012) Klein J (2012) Systems engineering framework for integrated product and industrial design including trade study optimization. In: Dagli CH (ed) New challenges in systems engineering and architecting conference on systems engineering research (CSER). Elsevier, Amsterdam, pp 413–429
58. Van Ruijven LC (2012) Ontology and model-based systems engineering. In: Dagli CH (ed) New challenges in systems engineering and architecting conference on systems engineering research (CSER). Elsevier, Amsterdam, pp 194–200
59. Boehm B, Koolmanojwong S, Lane JA, Turner R (2012) Principles for successful systems engineering. In: Dagli CH (ed) New challenges in systems engineering and architecting conference on systems engineering research (CSER). Elsevier, Amsterdam, pp 297–302
60. Boy GA, Narkevicius JMG (2014) Unifying human centered design and systems engineering for human systems integration. In: Aiguier M et al (eds) Complex systems design and management 2013. Springer International Publishing, Switzerland
61. Bartelt C, Böß V, Brüning J, Rausch A, Denkena B, Tatou JP (2013) A software architecture to synchronize interactivity of concurrent simulations in systems engineering. In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 19–29
62. Cole R (2009) System of systems architecture. In: Jamshidi M (ed) System of systems engineering: principles and applications. CRC Press, Boca Raton, pp 37–70
63. NN (2011) ISO/IEC/IEEE 42010:2011—systems and software engineering—architecture description. Iso.org. 2011-11-24. Accessed 15 Feb 2014
64. NN (2012) Enterprise architecture. An overwiew. http://isa.unomaha.edu/wp-content/uploads/2012/08/Enterprise-Architecture.pdf. Accessed 3 Aug 2014
65. Urbaczewski L, Mrdalj S (2006) A comparison of enterprise architecture frameworks. Issues Inf Syst 7(2):18–26
66. Reich C, Burghard O (2009) Architekturentwicklung in der wehrtechnischen Industrie. http://www.bitkom.org/files/documents/Leitfaden_ArchitekturentwicklungInDerWtIndustrie.pdf. Accessed 3 Aug 2014
67. Albers A, Zingel C (2013) Challenges of model-based systems engineering: a study towards unified term understanding and the state of usage of SysML. In: Abramovici M, Stark R (eds) Smart product engineering. Springer, Berlin, pp 83–92
68. Rodriguez-Priego E, García-Izquierdo FJ, Rubio AL (2010) Modeling issues: a survival guide for a non-expert modeler. In Petriu DC, Rouquette N, Haugen, Ø (eds) MODELS 2010, Part II, LNCS 6395. Springer, Berlin 2010, pp 361–375
69. Sun J, Hiekata K, Yamato H, Nakagaki N, Sugawara A (2014) Virtualization and automation of curved shell plates manufacturing plan design process for knowledge elicitation. Int J Agile Syst Manage 7(3/4):282–303
70. Königs SF, Beier G, Figge A, Stark R (2012) Traceability in systems engineering—review of industrial practices, state-of-the-art technologies and new research solutions. Adv Eng Inform 26(2012):924–940
71. Chang D, Chen CH (2014) Understanding the influence of customers on product innovation. Int J Agile Syst Manage 7(3/4):348–364

72. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manage 7(2):116–131
73. Vuillemin B, Croue N, Loembe S (2012) MBSE applied to an aerospace “force fighting” application, ERTS2 2012—embedded real time software and systems, Toulouse, 1–3 Feb 2012. http://www.erts2012.org/Site/0P2RUC89/TA-2.pdf. Accessed 4 Aug 2014
74. Mattice JJ (2005) Hubble space telescope systems engineering case study. Defense Acquisition University. https://acc.dau.mil/adl/en-US/37600/file/9105/Hubble%20Space%20Telescope%20SE%20Case%20Study%20-%20JJ%20Mattice.pdf. Accessed 4 Aug 2014
75. Friedman G, Sage AP (2004) Case studies of systems engineering and management in systems acquisition. Syst Eng 7(1):84–96
76. NN (2010) Code of PLM openness, ProSTEP iVip association, Darmstadt. http://www.prostep.org/en/cpo.html. Acessed 4 Aug 2014
77. Nicholds BA, Mo J (2014) Determining an action plan for manufacturing system improvement: the theory. Int J Agile Syst Manage 6(4):324–344
78. Nicholds BA, Mo J, Bridger S (2014) Determining an action plan for manufacturing system improvement: a case study. Int J Agile Syst Manage 7(1):1–25
79. ElMaraghy W, ElMaraghy H, Tomiyama T, Monostori L (2012) Complexity in engineering design and manufacturing. CIRP Ann Manufact Technol 61:793–814
80. Ito T (2014) A proposal of body movement-based interaction towards remote collaboration for concurrent engineering. Int J Agile Syst Manage 7(3/4):365–382
81. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manage 7(2):101–115
82. Moynihan P, Dai W (2011) Agile supply chain management: a services system approach. Int J Agile Syst Manage 4(4):280–300
83. Jacobs MA (2013) Complexity: toward an empirical measure. Technovation 33(2013):111–118
84. Carvalho H (2013) An innovative agile and resilient index for the automotive supply chain. Int J Agile Syst Manage 6(3):258–278
85. Modrak V, Semanco P (2012) Structural complexity assessment: a design and management tool for supply chain optimization. Procedia CIRP 3:227–232

Chapter 10
Knowledge-Based Engineering

Josip Stjepandić, Wim J.C. Verhagen, Harald Liese and Pablo Bermell-Garcia

Abstract The handling of knowledge represents the key to competitiveness, with company-specific product and process knowledge marking a unique position with respect to competition. Knowledge-based engineering (KBE) is a comprehensive application of artificial intelligence in engineering. It facilitates new product development by automating repetitive design tasks through acquisition, capture, transform, retention, share, and (re-)use of product and process knowledge. The idea behind KBE is to store engineering knowledge once by suitable, user friendly means and use it whenever necessary in a formal, well documented, repeatable and traceable process. It works like design automation. This chapter begins with the definition of knowledge in an engineering context and subsequently addresses the state-of-the-art in KBE research. Three particular areas of research are discussed in detail: knowledge structuring, maintainability of knowledge and KBE applications, and the technological progress and weaknesses of commercial KBE applications like KBE templates. From case study examples, various recent developments in KBE research, development and industrial exploitation are highlighted. By the resulting sequence optimization of the design process a significant time saving can be achieved. However, there are still notable drawbacks such as the complexity of KBE implementation and the adaptability of developed applications that need to be researched and solved. A view on KBE systems within the Concurrent Engineering context is synthesized, leading to the identification of future directions for research.

J. Stjepandić (✉) · H. Liese
PROSTEP AG, Dolivostr 11, 64293 Darmstadt, Germany
e-mail: josip.stjepandic@prostep.com

H. Liese
e-mail: harald.liese@prostep.com

W.J.C. Verhagen
Aerospace Engineering, Air Transport and Operations, TU Delft, 2600 AC Delft, Netherlands
e-mail: W.J.C.Verhagen@tudelft.nl

P. Bermell-Garcia
EADS Innovation Works, Filton, Bristol, UK
e-mail: Pablo.Bermell-Garcia@airbus.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_10

Keywords New product development · Virtual product creation · Artificial intelligence · KBE · Knowledge-based engineering · Knowledge management · Template

10.1 Introduction

During the development, production and operation of products, knowledge is created in various forms. From an engineering perspective, knowledge can be defined as processed information with a capability for effective action, where information can be seen as data within a structured context [1, 2]. Consequently, a hallmark of knowledge is that it is applied and sometimes revised or discarded in processes spanning the product lifecycle. The central role of knowledge in all aspects of the product lifecycle is well recognized, as is its function as a driver for the competitiveness of an organization through the capability for effective action [3]. Coupled with the increasing availability of affordable information technology, a proliferation of initiatives related to knowledge systems has developed over the years, both within and outside the engineering domain. These systems capitalize on various research streams to capture, structure and (re-)use product lifecycle knowledge within an organisation. Following the concurrent engineering idea, this product lifecycle knowledge should be systematically considered in the (early) design of new products. Increasingly, knowledge-based approaches are developed to facilitate simultaneous consideration of product lifecycle aspects.

One of the most promising application fields in this respect is knowledge-based engineering (KBE). The objective of KBE is to reduce time and costs of new product development, which is primarily achieved through automation of repetitive design tasks while capturing, retaining and re-using product and process knowledge [4]. In recent years, KBE methodologies, technologies and systems have been developed that facilitate new product development taking into account product lifecycle requirements, constraints and knowledge. In light of these developments, this contribution has two aims. First, research challenges relative to KBE from a lifecycle perspective are identified. Second, recent practical applications of KBE addressing these challenges are highlighted. In particular, the issues of capturing and structuring knowledge, maintaining knowledge over the life of KBE applications and integrating KBE functionality into CAD systems are discussed.

The structure of this chapter reflects these aims. In Sect. 10.2, KBE and its supporting principles, methodologies and technology are briefly introduced. Advances with respect to the capturing and structuring of knowledge are discussed in Sect. 10.3: Engineering Ontologies. Subsequently, research related to the issue of knowledge usability and maintenance in knowledge-based systems (KBSs) is introduced in Sect. 10.4. Recent research efforts have sought to introduce maintainable KBE functionality into CAD systems (Sect. 10.5). Section 10.6 showcases the results of developing industrial KBE applications for various industries, including the automotive, aircraft, and shipbuilding industries. It includes use cases for electrical engineering, dentistry, and manufacturing, too. A discussion chapter gives insight into benefits and gaps of current applications of KBE as well as future directives. Finally, an outlook is given with respect to the future of KBE from a CE perspective.

10.2 Knowledge-Based Engineering: Context, Principles and Challenges

To properly place KBE, its positioning relative to other knowledge-related research fields needs to be understood. La Rocca [5] positions KBE relative to knowledge management (KM) and knowledge engineering (KE).

Figure 10.1 shows this positioning together with associated knowledge technologies, as bullet-listed. KM is shown as the encompassing area, where the attention is on the overall goal of nurturing and supporting initiatives that can enable a more efficient and effective use of knowledge assets in the organisation [5]. The objectives of KM are to make knowledge visible and usable, to develop a 'knowledge-intensive culture' where knowledge is proactively shared, and to build a supporting knowledge infrastructure, including systems and people [2]. According to Alavi and Leidner [2], the basic processes involved in KM are creating, storing/retrieving, transferring and applying knowledge.

The research discipline of KE can be seen as a subset of KM. It focuses on the acquisition and codification of knowledge to support the development, implementation and maintenance of KBSs, which is supported by various methodologies, e.g., common knowledge acquisition and documentation structuring (CommonKADS),

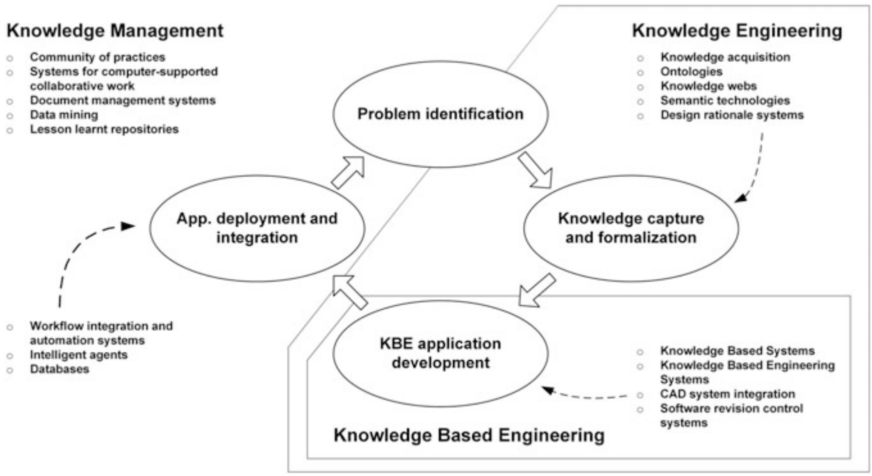


Fig. 10.1 Relative positioning of KBE versus KM and KE [5]

model-based and incremental knowledge engineering (MIKE), and Protégé-II [6]. KBSs are systems that use a formalized set of knowledge to offer problem-solving advice (also known as Expert Systems) or to solve problems directly. KBSs are typically comprised of acquisition mechanisms, a structured knowledge base containing a body of domain knowledge and reasoning mechanisms to solve the problems at hand. As such, KBSs typically comprise the two main elements used to formally represent knowledge: ontology and logic. An ontology can be defined as an explicit conceptualization of a domain, representing a structured view of the concepts and relationships in a domain—similar to the definition of *information* as data within a structured context. The concepts and relationships of an ontology are often expressed using formal language, such as predicate logic, which enables the construction of structured knowledge bases upon which logical operations (inferences) can be performed through reasoning mechanisms. Alternatively, the reasoning capability within KBSs relies on rule-based or object-oriented (OO) approaches. Most importantly, the reasoning capability offered by KBSs realizes the capability for effective action commonly associated with definitions of knowledge [2, 7, 8].

From a design engineering perspective, the most notable shortcomings of KBSs are that they lack a capability for geometry manipulation and data handling [4]. Ideally, the KBS capabilities regarding knowledge capture, knowledge representation and reasoning are to be merged with computer-aided design (CAD) and computer-aided analysis (CAA) capabilities to provide engineers with automated assistance in geometry manipulation, data processing and analysis. To achieve just this, KBE initiatives originated in the early 1980s.

KBE can be seen as an extension of KE/KBS into the design engineering domain, adding facilities for geometry manipulation and data handling capabilities [5]. KBE is characterised by its language-based, OO approach. KBE systems are used as general purpose tools to develop KBE applications through a programming approach using KBE programming languages. Using such an approach, the application of KBE to automate routine design tasks has demonstrably resulted in significant savings as well as more flexible and faster exploration of the design space [9, 10].

Though KBE originated in the early 1980s, the high costs of dedicated hardware, relatively esoteric programming languages and lack of methodological support and guidelines prevented widespread adoption beyond large automotive and aerospace OEMs [5]. In recent years however, advances have been made on all these fronts, enabling more companies and researchers to take up KBE and develop applications (see Sect. 10.6 for examples).

An important advance in KBE research has been made by the ESPRIT Framework IV Methodology and tools Oriented to Knowledge-based Applications (MOKA) project which ran from 1998 to 2002. The MOKA project was the first to identify a clear lifecycle for KBE systems and applications, consisting of eight steps. The KBE System Lifecycle is a process view of the lifecycle stages of a KBE system. The first stages (Identify; Justify) have the purpose to identify, analyse and scope opportunities for KBE development. The next stages are Capture and Formalise; their purpose is to capture and model the knowledge and activities that are associated with the KBE project. The remaining stages are Package, Distribute, Introduce and Use, which handle the implementation, distribution and use of KBE applications in an organisation.

Using the KBE System Lifecycle as a basis, the MOKA methodology was consequently developed to guide the development of KBE applications by taking a project from inception towards industrialization and actual use [11]. The Capture and Formalise stages are the central contribution of the MOKA methodology. The centrepieces of these stages are the Informal and Formal MOKA models. The informal model consists of so-called ICARE forms, where the acronym stands for Illustrations, Constraints, Activities, Rules and Entities. These forms can be used to decompose and store individual knowledge elements. Subsequently, these elements can be linked to create a structured web of knowledge elements that together make up a representation of the problem domain to which users from multiple viewpoints can relate. When the problem knowledge has been converted into a structured representation, the next step is to formalize this knowledge in order to represent knowledge in a form that is acceptable to knowledge and software engineers and suitable for subsequent development of a KBE application. The formal model uses [Moka Modelling Language (MML), an adaptation of Unified Modelling Language (UML)] to classify and structure the ICARE informal model elements, which are translated into formal Product and Process models. The main elements of the MOKA methodology are illustrated in Fig. 10.2: the KBE system lifecycle, the Informal model (as illustrated by an ICARE form) and the Formal model (as represented by an MML structure).

Recent research on KBE focuses on resolving current KBE challenges (e.g., black-box implementation and (non-)transparency of knowledge bases, ad hoc development of KBE applications, quantification of KBE opportunities as well as results [9] and integration of KBE with other major research streams and themes, for instance multi-disciplinary optimization (MDO), systems engineering (SE) [12, 13], and Product Lifecycle Management [14]. In this contribution, the following three research issues are discussed into more depth:

- **Structuring knowledge**: in recent years, the development of ontologies has become a strong supporting element in KBE and PLM development [12, 15]. It is imperative to develop formally structured yet transparent knowledge bases. To this end, a close coupling is desired between KBE applications and the supporting knowledge base(s) developed using principles from ontology engineering and KM. Even more ambitiously, supporting transitions from knowledge base towards KBE software code and vice versa is of particular relevance, as advocated by the iPROD project features on automatic code generation and round-tripping [12]. In Sect. 10.3, recent work on engineering ontologies for KBE applications is discussed.

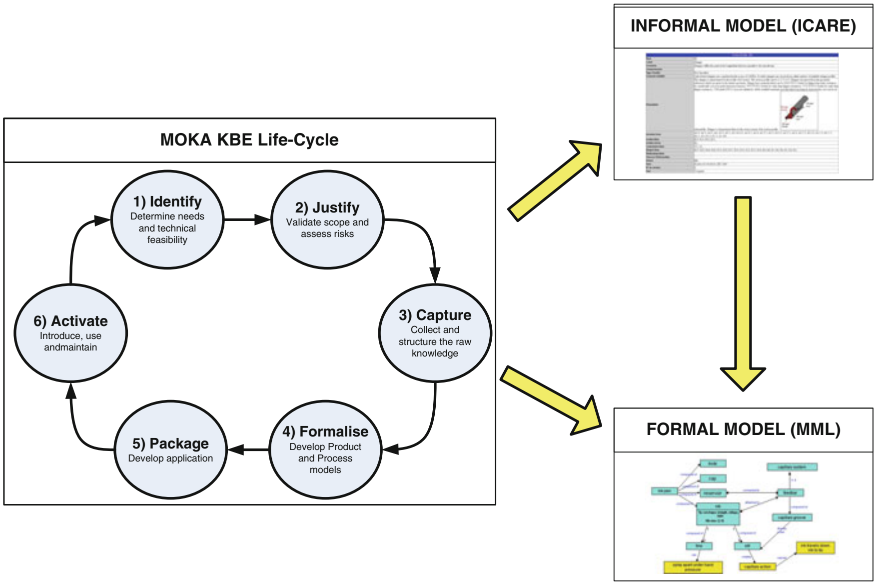


Fig. 10.2 MOKA methodology elements [11]

- **Maintainability and usability of knowledge through life**: during product life, knowledge is not only created but also tends to change over time. This creates a requirement to keep a knowledge-based application usable over time by revising, expanding or contracting the knowledge base. Improving the maintainability of KBE applications is necessary to improve the long-term viability of these applications. Though maintenance is explicitly identified in MOKA, it does not offer guidelines for maintenance of KBE systems or applications other than to repeat the whole KBE System Lifecycle process when changes in knowledge occur. Furthermore, it is not clear how existing models built in the Capture and Formalise stages are to be adapted given new or changed knowledge. Moreover, it is not clear how these changes are to be propagated into the packaged KBE system while maintaining knowledge base consistency and reliability. In Sect. 10.4, the usability and maintainability of KBE applications is discussed in more detail.
- **KBE functionality in CAD systems**: commercial CAD vendors are increasingly offering 'KBE-like' functionality as part of their products. Though the supporting technology is different in essence [5], the result is similar: through KBE templates in CAD systems, engineering tasks can be automated, yielding significant savings while capturing, storing and re-using product and process knowledge. In Sect. 10.5, research into developments on this front is presented.

10.3 Engineering Ontologies

An ontology can be defined as a definition of a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and the relations among them. An ontology necessarily includes a common vocabulary of terms and a specification of their meaning [16]. Without specification, the set of ontology concepts would be variously interpretable by different sets of users. With specification, different users (e.g., experts in different lifecycle phases) with different views on a single reality can be accommodated.

Ontologies can be built using various methodologies (e.g., METHONTOLOGY [16]). When properly developed and implemented, ontologies can serve as the backbone for knowledge-based applications. They offer the possibility to structure the knowledge base by modelling the context in which knowledge is viewed. Domain (meta)models are made explicit and knowledge (re-)use is made possible [17]. Thus, as mentioned, ontologies can incorporate the use of predicates and an inference capability, which offers the potential to execute automated reasoning upon the knowledge base. Finally, ontologies are flexible and can be extended. As such, ontologies not only support multiple viewpoints on the same knowledge, but also offer critical functionality for knowledge-based applications.

Ontologies can vary across multiple aspects, the foremost of which are the degree of formality (from highly informal to rigorously formal), purpose, and subject matter [16]. With respect to the latter, Uschold and Gruninger [16] identify three main categories, namely domain ontologies, task/problem solving ontologies, and meta-ontologies. In combination, tasks (reasoning process) and domain knowledge (facts about a domain) can be used in KBSs to automate activities. With respect to task ontologies, Kitamura et al. [18] present an ontological framework to systematically describe and deploy functional knowledge. Other task ontologies are discussed as part of the literature on problem-solving methods (PSM) [19].

With respect to domain ontologies, various authors propose ontology-based approaches to support Product Data Management and/or Lifecycle Management [17, 20]. When considering commercial models, various engineering applications developed by Dassault Systèmes (e.g., Catia™ and Delmia™) use the Product-Process-Resource (PPR) model. As Curran et al. [21] note, this model separates product development into the three domains of PPR, enabling the construction of OO tree structures capable of modeling the hierarchies of and all logical relationships between the process, product and resource data.

From a KBE perspective, MOKA also takes into account activities as part of its ICARE ontology, but this cannot be considered to be a full-fledged task ontology onto itself. As a response, Vermeulen [15] presents an ontology for solution-finding strategies with respect to engineering tasks. Ruschitzka et al. [22] propose an ontology-based approach for stamping-die design in which a task perspective is central.

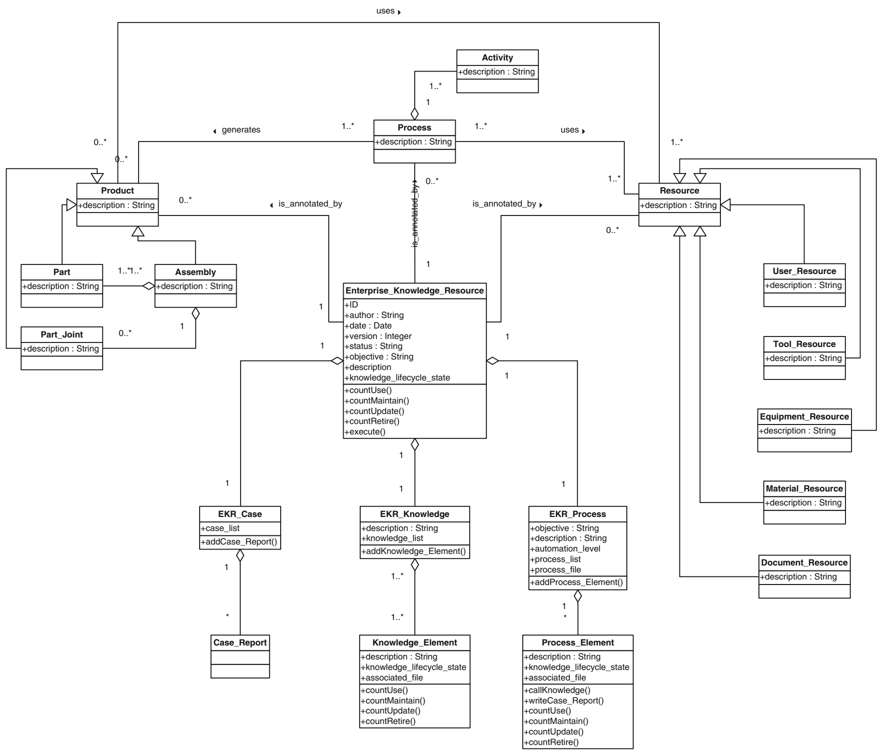


Fig. 10.3 Knowledge lifecycle ontology [7]

In recent years, the notion of constructing task and domain ontologies to support KBE applications has gained more traction. In the iPROD project [12], there is some emphasis on the construction of task ontologies for generic engineering tasks (e.g., optimization, design) and mapping these onto domain-specific problems. iPROD also uses ontologies to facilitate automatic code generation and round-tripping between knowledge base and KBE application.

In work by Bermell-Garcia et al. [23] and Verhagen [7], an ontology-based approach is adopted to model and automate engineering tasks while using domain-specific knowledge. Figure 10.3 shows the ontology developed for this approach: the Knowledge Lifecycle Ontology. The Knowledge Lifecycle Ontology consists of two central perspectives:

- **Enterprise Knowledge Resource** (**EKR**): a task-oriented container of the knowledge, subtasks and output associated with an engineering task. In other words, an EKR can be used to represent the content of a task.
- **PPR**: classes to represent the product(s), process(es) and resource(s) an EKR is associated with. In other words, these classes represent the context of a task.

As the task is separated from the domain knowledge, it is possible to independently update either element. Given the association of the task with its context, it becomes easier to retrieve and access a task as well as its individual elements. In short, maintainability and usability of a knowledge-based application are improved. This issue is discussed in more detail in the following Section.

10.4 Knowledge-Based Applications Through Life

Knowledge tends to change over time [4]. Knowledge change can be defined as a change in knowledge over time, where knowledge is defined as processed information with a capability for effective action. Modes of change may include data change (values associated with knowledge elements alter from time t_1 to time t_2), information change (the structured context of a knowledge element changes) and knowledge change (the capability for effective action associated with a knowledge element can change, caused by changes in rules, logic or attribute sets, depending on the formalism chosen to represent knowledge).

When knowledge used in a knowledge-based application changes, the application has to be maintained. The associated maintenance may impact the majority of a knowledge base as well as dictate a large part of its total cost of ownership (up to 25 % of initial investment on a yearly basis, as indicated in Van Dijk et al. [13]). Improving the maintainability of KBE applications is consequently necessary to improve the long-term viability and usability of these applications.

As discussed in Verhagen et al. [24] and Verhagen [7], maintainability of KBE applications necessitates transparent KBE applications that move beyond 'black-box': all too often, the knowledge contained in the KBE applications is difficult to access and inspect, and is often embedded in the application code [24]. To improve it is necessary to support categorization, accessibility and traceability of knowledge. To achieve transparency, knowledge included into knowledge-based applications should be substantiated: the underlying knowledge and supporting documentation for the knowledge-based application should be categorized and be directly accessible. To enable effective use and update of knowledge, the knowledge element(s) in applications should be formally structured using knowledge model(s) and metadata.

The end goal of KBE maintenance is to ensure usable knowledge-based applications. This can be facilitated through a task orientation and expert and/or end user involvement. With respect to a task orientation, knowledge involves a capability for effective action. This capability can be met by explicitly associating sets of knowledge with functional tasks. Furthermore, end users must be able to identify, use, interact with and if necessary, update the relevant knowledge that they use in their daily workflow and specific context. This requires that knowledge is tied to engineering tasks and that knowledge is visible and directly accessible for end users to enable interaction—context and semantics must be provided.

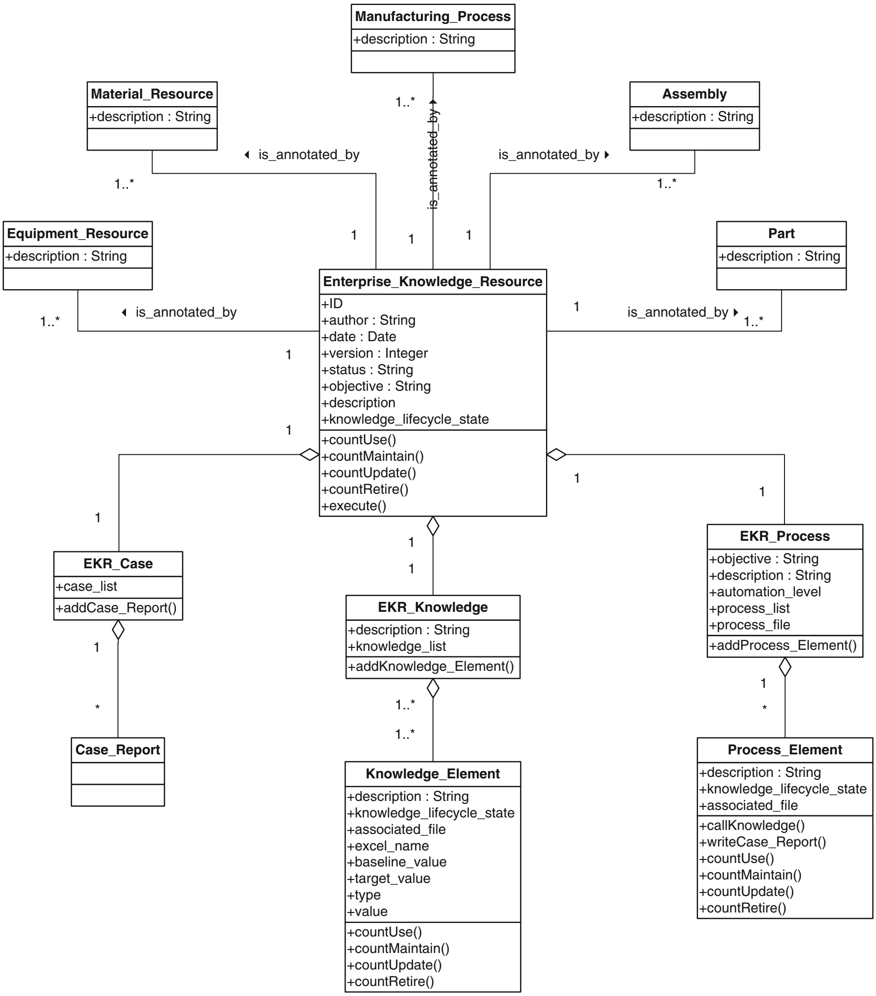


Fig. 10.4 Knowledge lifecycle ontology applied to manufacturing cost modelling task [7]

The Knowledge Lifecycle Ontology introduced in Sect. 10.3 has been configured to allow for maintainable and usable KBE applications through life: it provides a structural representation for tasks as well as individual knowledge elements, enabling the development of transparent applications. The ontology includes classes to model task context and provide semantic annotation, leading to improved visibility and accessibility of engineering tasks.

The ontology has been applied in the development of several knowledge-based applications that automate engineering tasks. Examples include manufacturing cost modelling and estimation for composite wing covers [24] and ply stacking sequence optimization for composite wing covers [9]. Figure 10.4 shows the application of

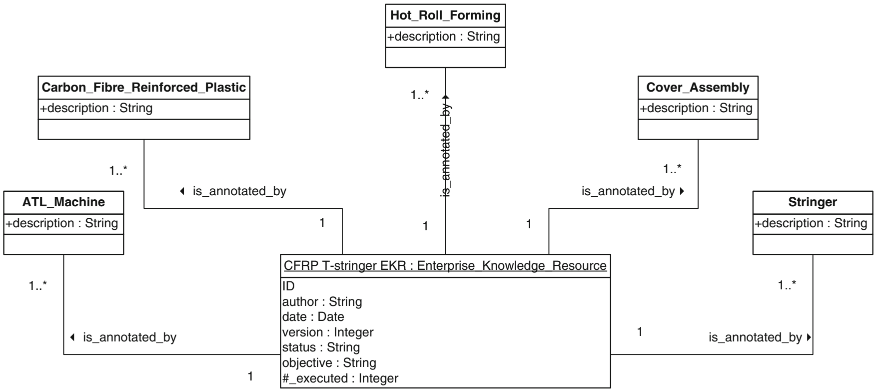


Fig. 10.5 Instantiated enterprise knowledge resource [7]

the ontology for the manufacturing cost modelling task; the developed EKR is shown along with the semantic annotation for this resource. Figure 10.5 shows an example of an instantiated EKR.

10.5 KBE Applications and Templates

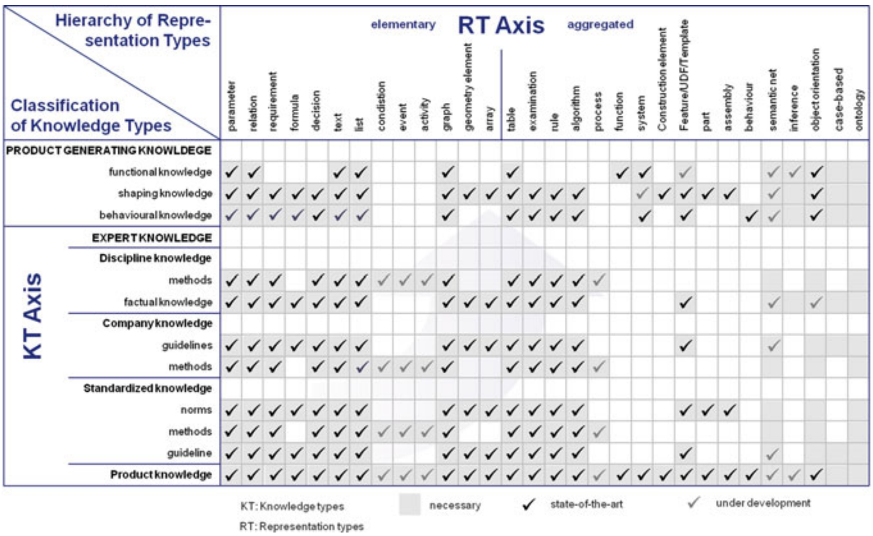
The use of KM occurs primarily with CAD systems in two ways [25]. First, there is an extended programming interface with the functionality of an AI language. So it is possible to create own KBE applications (e.g., a module for die design) like any other CAD application in a software development process. Thus, expert knowledge of software development is needed which most CAD users don't have. As this approach is very complicated and risky for most enterprises, many additional commercial KBE modules are offered, in which KBE templates can be created interactively in software development without expert knowledge. Furthermore, both approaches can be combined with each other and with the application programming interface (API) additional tools can be integrated in a KBE process. Both the KBE application and the KBE template are part of the software lifecycle and have to be maintained in case of a release change like any other user software [6], as shown in Sect. 10.4. Contrary to a KBE application which consists of an executable software code that has to be installed at every workstation, a KBE template consists of one or more CAD models which have to be instantiated by use. Simplifying, in the following discussion we are talking only about KBE templates.

10.5.1 General Representation

The procedure for the creation of knowledge-based 3D-CAD models is based on multifaceted types of representation which are implemented in leading CAD systems depending on the specific forms of knowledge (Fig. 10.6) and covers possibilities for action of the product designer in the knowledge-based model. The procedure consists of individually associated methods and operates based on the developed generic representation model (Fig. 10.7).

The arrows in Fig. 10.7 describe abstract sequences, where two different semantics of sequences exist. On the one hand, it is possible to use mechanisms within the representation (which e.g., ensure proper knowledge integration of new elements into a CAD model), whereas other arrows describe the derivation of the presentation out of the representation. In the knowledge representation arrows named “mapping” (1–2) connect methods with formalized knowledge and the knowledge definition. The methods are located inside the formalized knowledge representation, because they are a part of an OO representation and are associated with the respective context. They are related to basic methods of modeling and to elementary methods of KM.

The main activity in the knowledge-based 3D-CAD design or modeling process is the use of different types of knowledge representations (3). In this case, selected patterns or templates provided by the 3D-CAD system are to be integrated in the 3D-CAD model (4), where they are also instantiated. Through instantiating, new objects in the CAD model or new CAD models (e.g., templates) are created.



RT Axis: elementary → aggregated; KT Axis

Hierarchy of Representation Types / Classification of Knowledge Types	parameter	relation	requirement	formula	decision	text	list	condistion	event	activity	graph	geometry element	array	table	examination	rule	algorithm	process	function	system	Construction element	Feature/UDF/Template	part	assembly	behaviour	semantic net	inference	object orientation	case-based	ontology
PRODUCT GENERATING KNOWLDEGE																														
functional knowledge	✓	✓				✓	✓				✓			✓					✓	✓		(✓)				(✓)	(✓)	✓		
shaping knowledge	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓			(✓)	✓	✓	✓	✓		(✓)		✓		
behavioural knowledge	✓	✓	✓	✓	✓	✓	✓				✓			✓	✓	✓	✓			✓		✓			✓	(✓)		✓		
EXPERT KNOWLEDGE																														
Discipline knowledge																														
methods	✓	✓	✓		✓	✓	✓	(✓)	(✓)	(✓)	✓			✓	✓	✓	✓	(✓)												
factual knowledge	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓					✓				(✓)		(✓)		
Company knowledge																														
guidelines	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓					✓				(✓)				
methods	✓	✓	✓		✓	✓	✓	(✓)	(✓)	(✓)	✓			✓	✓	✓	✓	(✓)												
Standardized knowledge																														
norms	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓					✓	✓	✓						
methods	✓	✓	✓		✓	✓	✓	(✓)	(✓)	(✓)	✓			✓	✓	✓	✓	(✓)												
guideline	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓					✓				(✓)				
Product knowledge	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	✓	✓	✓	✓	✓	✓	✓	(✓)	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	✓		

KT: Knowledge types; RT: Representation types; [grey field] necessary; ✓ state-of-the-art; (✓) [grey check] under development

Fig. 10.6 Overview of knowledge types versus representation types [4]

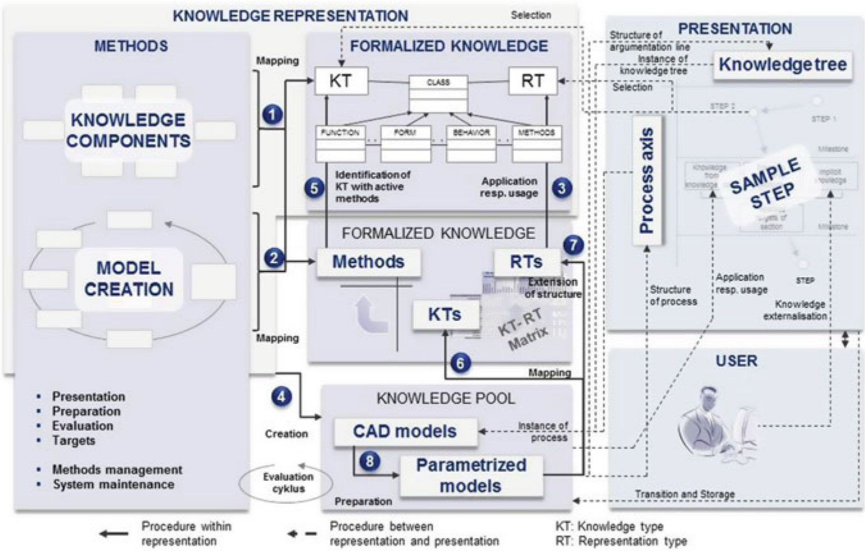


Fig. 10.7 3D Knowledge representation model with relationships [4]

If a type of representation can be directly mapped to a specific type of knowledge, it is possible that an instant allocation of knowledge and knowledge methods occurs. When the user is applying a knowledge module, the type of knowledge can be identified based on the module chosen by the user and thus additional types of knowledge representations respectively knowledge structures can be determined that are capable to represent the chosen knowledge type. Methods should be provided to the user, which specifically support knowledge handling. If one of these methods is chosen and if the one is clearly dedicated to a type of knowledge (context), an identification of the associated type of knowledge over activated methods can be effected (5). If a type of knowledge is identified, which represents the normal case, an object should be generated for this type of knowledge within the representation. At the end of the 3D CAD design process, based on these operations many instantiated representation and knowledge objects should exist. A simple example for providing knowledge representations due to the context is the module/workbench concept of commercial CAD systems. Successive to the user choices special representations or feature types are provided by the module to represent, i.e. engineering domain knowledge. Later on another user or a system can even inference the knowledge types in the CAD model based on the knowledge representations used—if a 1:1 association between knowledge representations and knowledge types is possible. A more enhanced example is the automated use of templates in the sequence of a business process template. Here based on the user interactions and context, different templates with knowledge representations can be used in the modeling process, so that at the end of the 3D CAD design process subsequently a number of instantiated temples compose the knowledge base. If reasonable—from such a knowledge pool new superior representation types can be extracted and assigned to certain knowledge types.

The knowledge pool is linked with the definition of knowledge in such a way, that an extension of knowledge types (6) and representation types (7) out of the parameterized 3D CAD models is possible. In case of generation of new parametric models (8) they compose structures or even templates, combining existing types of representations into a new one. Thereby, this new sample has to find its integration into the existing hierarchy of representation types. This assumes an assignment of the composed knowledge representation to one or multiple types of knowledge. This method of extension of the types of representation supports directly one of the most fundamentals of the AI approach of the knowledge representation formalism. This implies that newcomers use declarative knowledge structures whereas experts manage their knowledge in procedural form as chunks, compact knowledge units on a higher abstraction level. The 3-D CAD models as well as the parametric models are generated from building blocks taken from the knowledge representation. Parametric models can be generated by the rework of conventional 3-D CAD models as has been described above. CAD models become "intelligent" including knowledge objects in addition to geometry and features.

The knowledge presentation has to be derived and visualized in a suitable form. Partly, this occurs through the conventional description of the relations and design elements of the 3-D CAD system in the form of a structure or model tree. However, there is a necessity to present the knowledge in the context of processes, like the design process or the modeling process. Initial approaches exist for process presentation in the various CAD systems with a huge potential for development.

10.5.2 KBE Templates Hierarchy

KBE templates as typical applications of KBE are a comprehensive means for acquiring, processing and managing essential information and knowledge in a standardized product and process definition [25, 26]. They are developed with a specified procedure: A specified operation or method is mapped by the CAD system with the support of representation types from different CAD modules [25]. In Fig. 10.8, a CAD-based template hierarchy is presented.

From the top to the bottom of the template hierarchy the scope and the content of the respective templates increase considerably, e.g., from simple geometry to complex system representations. Furthermore also the complexity of the creation of the template and the required representation mechanisms increases rapidly for the template creator. One important challenge for end-user acceptance and to master the increasing and challenging template contents is to package the complexity for the end user in an easy-to use black box or grey box with defined interfaces to the other CAD elements at template instantiation.

For industrial practice part/geometry templates, product/assembly templates, knowledge templates, behavior templates, test/validation templates, process chain

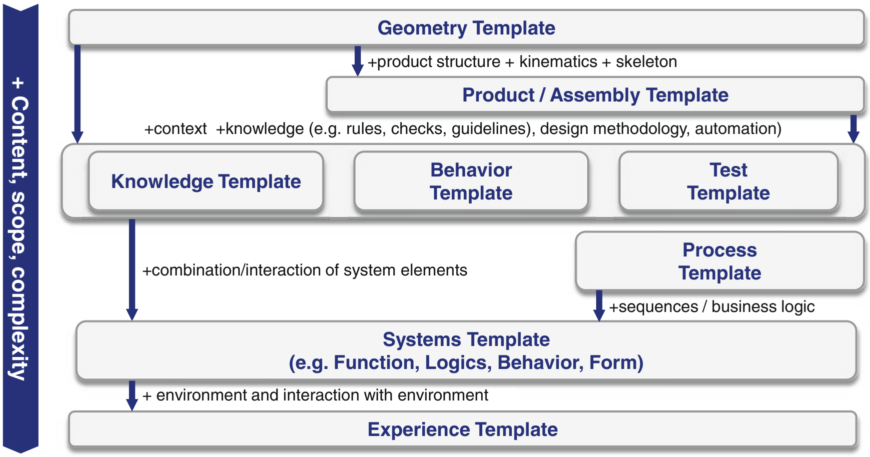


Fig. 10.8 KBE templates hierarchy [4]

templates and system templates matter. Part/geometry templates contain geometry or features for easy re-use. Product/assembly templates contain geometry as well as product structures with sub-assemblies and special assembly elements like e.g., kinematics definitions. Also design methodology can be represented, e.g., a skeleton method to represent a top-down design method. Template types can be extended with information about the template context and special discipline knowledge (e.g., rules, checks and guidelines) to become knowledge, behavior or test templates. They can contain design methodologies or design rules as well as automation capabilities that can be used for automatic parts and sub-assembly generation. Often, knowledge features are used in connection with skeleton model methods which describe references and parameters that are necessary to shape and instantiate parts in a sub-assembly. Validation templates serve for the testing of complex rules (e.g., legitimate duties) and contain validation procedures or validation features. Behavior templates contain elements used to support simulation aspects of the product definition, e.g., CAE templates for DMU or FEM simulation, which typically are fully associative with geometry and parametric elements connected to the template. A process template represents a structured sequence of tasks related to the CAD-system elements based on the underlying business process. A CAD-based process template controls the information flow between the CAD elements and triggers multiple actions based on user interaction or dedicated events, e.g., instantiation of other templates. A process chain template extends this scope to connect different domains (e.g., design to manufacturing). System templates combine all preceding elements listed in the template hierarchy in order to represent complex building blocks of a system. Thus in a systems template functions, logics, behavior and form/geometry as well as the interaction between these system elements can be combined, e.g., to connect descriptions of functions and behavior with the shape. The connections of system description to requirements, which are typically represented in an external requirement management tool or in the PDM System, even enrich the context of the systems template. Currently systems templates represent the maximum state of the art in CAD systems, while many representation and linkage types as well as template methodologies have to be further investigated.

The concept of experience templates is an outlook onto the future. CAD and PLM Systems are more and more developed to integrate product definitions with models of the environment with which the product will be interacting. By the time the corresponding representations will be state of the art, the demand for re-usable template-based structures will emerge.

All kinds of templates encompass interface to embed external software to integrate processes or types of representation which is not provided by the KBE modules of the CAD systems. Different template types can be combined with each other in a work package.

10.6 Applications and Use Cases

KBE technology has become accepted in many industries. Especially, in use cases of design with a high percentage of repetition (see Chap. 14: Sects. 14.5.2 and 14.5.3) KBE technology offers massive benefits. Therefore, the name design automation is used too. Thereby, CAD data more and more become a storage for product and method knowledge, which can be recalled by need. In general, the benefit of KBE is extra high if products feature a high variance of a basis development, as much as possible single steps of the engineering process can be described by algorithms, and operations are cross-divisional (e.g., product design or tool design). By using a KBE template routine tasks are transformed into an automated process with a minimum of user assistance. The high one-time effort for the development and introduction of a template loses importance the more the template is used.

Basically, a methodical approach is premised in engineering processes, where job content consists of both creative and routine components. For an analysis phase, primarily the latter ones apply and need to be evaluated. A specification (rule base) occurs as a result followed by the generation of a KBE template. Finally in the third step, the KBE template is piloted, rolled out and deployed in practice like any other software. In the context of Concurrent Engineering, the synchronous development of jigs and tools for complete part families is, for example, imaginable, where repetition frequency and degree of phasing with product design are exceptionally high. With a KBE template parametric jigs and tool geometry can be automatically derived from the component geometry. The designer is led through the design process by means of templates and calculation rules integrated in scripts. By input of chosen parameter values and a reference geometry the desired appliance is designed (without more interaction). Here, a saving of time for the design of the appliance with a factor between 7 and 10 is achieved as proven during the introduction of KBE in industrial enterprises [25, 27].

10.6.1 Automotive

The automotive industry was, alongside the aerospace industry, the main driver of KBE (see Chap. 21) due to product complexity and variety, short product cycles and high investment volumes, where each new technology promises a quick return of investment. According to the corresponding phases in product development, function, concept, study and part templates can be distinguished [26]. Function templates contain only rough geometrical information and are mainly used for providing the main dimensions and specification features. An application of concept templates includes the main characteristics of vehicle models like sedan, convertible, station wagon or SUV. Digital validation of functional principles is the task of study templates. The detailing of such a validated concept is done in part (or assembly) templates. Templates will be explained below.

10.6.1.1 Part/Assembly Template

ProcedoStudio, a general tool (Fig. 10.9), has been developed for automatic variant design of car seat heaters based on varying customer specifications and seat geometries to be used in the quotation process [28]. Approximately 75 new variants are designed on a yearly basis and hundreds of requests for quotations are replied. Accuracy in quotations and short quotation preparation lead-time are key success factors for a company regarding quotations.

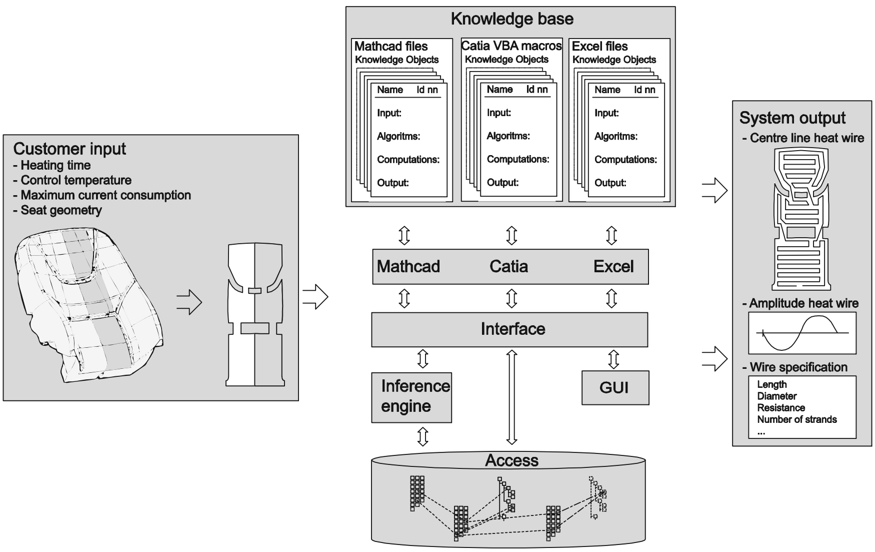


Fig. 10.9 Principle system architecture for automatic design of seat heaters [28]

The system is fed with customer-specific input (parameters with associated values together with a 2-D outline of the heating element). The main output includes a pattern for the heating wire's centre line, an amplitude factor for the sinusoidal loops, the wire specification, detailed manufacturing preparation and a cost estimation. The application for car seat heaters corresponds to a knowledge domain modeled as a Knowledge Base in the tool. Presently, there are 20 Knowledge Objects for input specification, file management, electrical calculations, geometry design, manufacturing preparation, and cost estimation. The number of variables managed by the database is 66, although the total number of variables residing in all of the Knowledge Objects is of much higher figure. Application programs used are MS Access 2007, MS Excel 2007, MathCAD 13, and CATIA V5R18.

The highly automated design of interior components and car brakes is introduced in industrial practice using a similar approach based either on module knowledge fusion (KF) in Siemens PLM NX [29] or workbenches Knowledgeware in CATIA V5 [30]. Apart from significant time savings, which justify the invest, high robustness in the design process and long-term knowledge preservation are achieved, too.

10.6.1.2 Process Chain Template

Die design is another typical use case for KBE caused by its crucial influence on project schedule and quality of body-in-white. During the process planning the number of dies are determined as well as the content of each die, including stamping direction, tasks and cam type. A study by You et al. combines practical experience with an expert system, and focuses mainly on pre-process steps and process planning in a study [31]. The computer-aided process planning system (CAPP) has been developed in Java. It uses the Spring Solid System as the CAD backbone to read the digital surface model, and then output the die layout using Java3D.

In process planning of automotive panels the number of operations, the tasks in each operation and the content of each task are identified. The CAPP system runs an automatic reasoning procedure. The essential tasks based on feature recognition and drawing direction results are searched and reasoned and then arranged for each operation. The most suitable stamping direction of each operation is analyzed, while the machining direction of each task is based on the stamping direction. The purpose of feature recognition is to categorize a panel into different sections to establish a bridge between a CAD model and the CAPP system because a panel model without feature recognition is merely unsorted surface data.

Automotive panels can be classified into appearance parts and structure parts. Appearance parts can be seen after assembly, including door, hood, fender, roof, and trunk lid; however, structure parts cannot be seen after assembly. This study uses the left side of a fender to illustrate the die layout design process, including feature recognition, machining center searching, drawing direction optimization, and process planning.

The operating time of this use case has taken about twenty minutes, and it will change with the file size and panel shape. The actual cost time except die face design from the experience of engineers on the factory is taken about 3–5 days.

The process planning result is not unique, and as such results will vary depending on specifications from different die design companies and different designers. If a die design company and its designers are the same, even when the panel is the same, process planning results will also vary due to different constraints (client demands and cost considerations). However, this application allows users to modify the content of process manually in response to different conditions and circumstances based on the acceptable and enforceable process. A similar practical approach based on hybrid knowledge representation with knowledge representation system (KRS) and CATIA V5 Knowledgeware is shown in Ruschitzka et al. [22].

10.6.1.3 Validation Template

Since the license for a passenger car is subject to many international rules, norms and standards, the KBE template CAVA (CATIA V5 Automotive Extensions Vehicle Architecture) was developed by Audi, BMW, Daimler, Porsche and Volkswagen to ensure car design and legal compliance during the entire design process—from concept phase to homologation [32].

As an additional CATIA V5 workbench CAVA creates reference or support geometry representing design space, clearance areas, or fields of vision required to support draft and design. Since this support geometry is generated from established standards, any existing or new geometry can be verified against these standards. The inserted references are fully associative, i.e. changes in data input prompt an update of the CAVA data. When non-compliance occurs, the user receives relevant advice. During the concept phase CAVA provides the boundaries for several design aspects and following the automatic check for legal conformity and reports deviations. Finally it validates that standards have been followed and creates reports to be used for homologation of a car.

CAVA includes a complete set of validation procedures like rear view mirror, viewing fields, security belts, underfloor clearances, lamp positions, pedestrian protection, and much more. Each function contains different parameters and settings that are determined by local authorities and described in corresponding norms.

Another KBE application ("Automatic Cross-Section Generator") which supports highly automated digital validation in the concept phase of car development based on modular architecture is described in Brüning and Liese [33] (see Sect. 13.6.2). It generates cross sections of all conceptual DMU data fully automatically primarily during the night and facilitates complex classification and validation procedures in batch modus comparing the working concept with other current and past product families.

10.6.2 Aerospace

Like the automotive industry aerospace is predestined for use of KBE for an additional reason: the long product cycles of airplanes require long-term knowledge capture in an enterprise preventing the loss of know-how by staff fluctuation after a project is finished. This is disclosed by two use cases.

10.6.2.1 Part/Assembly Template

A KBE application that supports the definition of the firtree (the contact interface between turbine discs and blades of low pressure turbines in an aircraft engine) is presented in Ugarte and Izaguirre [34]. Replacing an existing solution based on obsolete software technology, a new firtree application in the KBE language KF was developed for full integration into CAD system NX. It had to cover static stress check, developing the solid features for the disc in detailed and de-featured for FEA states, and developing the solid features for the blade in detailed, de-featured for FEA and casting states.

The application is implemented as a library of many KF classes comprising singular tasks. A class linked to the main dialog is used to mainly gather the desired state of the firtree and the profile positioning parameters with its attributes which are directly accessible in any other object generated by itself or any descendant, due to the automatic inheritance of attributes in KF. These parameters are passed to the Geometry Solver by a reference chain to the main class.

Thus, all parameters are available for the Geometry Solver class that analytically solves the detailed profile points. The Profile class supports the engineer during the preliminary geometrical design checks and simply draws the solutions with curves on the XY plane. The Static Stress class generates the profiles as sheet bodies to get the area of the profiles and performs the static stress calculations to aid the engineer during the preliminary checks. The Detailed Solid class positions the profile on the blade or wheel and extrudes the corresponding profile to generate the corresponding solid for the following Boolean operation. The CAE Solid class derives the truncations out of the inherited solution and again generates the solid by extruding the corresponding profile. Finally, the Cast Solid is only applicable to the blade and calculates the outmost lobe to add stock at a distance, followed by recalculation of a brand new cast profile.

For intermediate solutions which are not covered by the knowledge base stored in KF classes a procedure with user defined features (UDF) is defined enabling the users to choose between a set of firtree UDFs during the definition phase. The benefit of such an approach is the ability to quickly partially maintain the KBE application. The counterside of such a solution is that there is no support for the following states of the firtree.

10.6.2.2 Process Chain Template

In the aerospace domain, the introduction of lifecycle constraints into the design phase is a major theme, as encompassed by the Design for X philosophy. A major representation of this is Design for Manufacturing. Bermell-Garcia et al. [23] have developed a knowledge-based application to optimize of a composite wing cover conceptual design for ply continuity through the blending of stacking sequences, as explained below. The optimization of ply continuity in aircraft composite wing conceptual design is an example of addressing manufacturing considerations in an early stage of design. The industrial partner that participated in the study used a grid representation in the conceptual design of a carbon fibre reinforced plastic (CFRP) wing. In each of the grid cells, the amount of carbon fibre plies and their orientation is specified, based on structural requirements (minimum thickness and load cases). This specification, known as "ply stacking sequence", describes a particular sequence of composite layers, each of which has a specific fibre orientation. However, when considering the full grid, there can be significant mismatches between adjacent cell fibre orientations. The resulting discontinuities can be solved in manufacturing by introducing overlap and interleaving the adjacent layers. Plies are extended over the rib area and 'stacked' on top of each other, which introduces additional thickness (and mass) at the rib area, and consequently a ramp gradient from the rib to the cell. This ramp gradient must be kept within a specified limit depending on maximum tool deflection to ensure manufacturability. However, this solution adds considerable mass.

Instead of introducing additional mass through overlap and interleaving during manufacturing, it is preferable to reconfigure adjacent stacking sequences beforehand such that ply continuity is optimal and mass addition is minimal, while respecting the structural design requirements. In other words, if manufacturing considerations could be integrated into structural design and sizing, material and therefore mass, which is added later for manufacturing purposes, can be reduced. The preferred solution for the ply continuity problem comes down to re-sequencing and optimizing a set of stacking sequences such that ply continuity is maximized and minimum addition of mass is achieved, while obeying structural design requirements.

A knowledge-based application has been developed to solve the ply optimization problem using an automated solution, while retaining the required product and process knowledge [23]. This knowledge includes over 30 design and manufacturing constraints as well as pre-existing visual basic (VBA) code for ply reconfiguration. To support the solution, the Knowledge Lifecycle Ontology introduced in Sect. 10.3 has been applied as the semantic backbone. A task-oriented model known as an EKR has been developed to represent the knowledge, process and case output of the engineering task (ply optimization), augmented by a PPR metamodel for annotation of the engineering task and its components. Furthermore, a genetic algorithm has been implemented in Fortran to perform the optimization of ply continuity. To implement the solution, two main architectural elements have been devised:

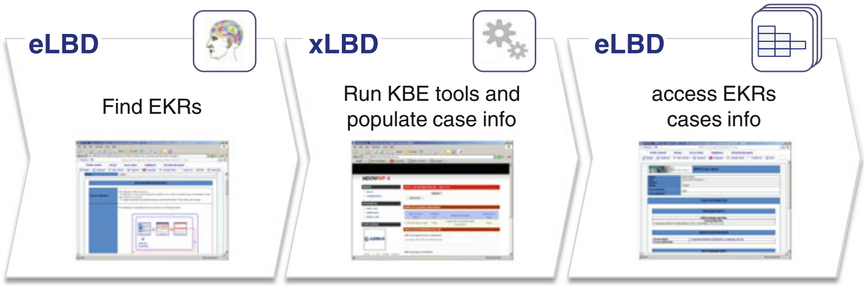


Fig. 10.10 Architecture for wing composite cover ply optimization EKR

- **EKR Environment for Learning by Doing** (**eLBD**): The eLBD is a web solution aimed at supporting end users. eLBD is based on a KM tool called Ardans Knowledge Maker (AKM). Specific models for the representation of knowledge and process elements have been implemented within AKM to enable the construction of EKRs which package the process and knowledge elements and the cases. The role of eLBD is not to store concept data but the collective thought behind the data (assumptions, constraints, rules, procedures and tools).
- **Executable environment for Learning by Doing** (**xLBD**): The xLBD is a solution to enable the remote execution of EKRs through a web service approach. xLBD uses several software applications and languages (Apache Tomcat web server, Java, AKM web services and Phoenix Integration Model Center®) to deploy EKRs as web services. Users can access and execute the software remotely, so they do not require a dedicated installation of software on their desktops.

The knowledge framework, with the eLBD and xLBD environments at its core, is shown in Fig. 10.10. The eLBD environment allows users to access EKRs and their constituent elements, whereas the xLBD environment allows (remote) execution of an EKR.

The implemented knowledge-based solution is able to deliver composite wing conceptual designs optimized for manufacturing. Evaluation and trade-off of the conceptual designs is supported by Pareto fronts using weight, cost and manufacturability objectives. The solution is currently used in industrial practice.

10.6.3 Space

The first design of a satellite for a specific mission is based on the main criteria payload and orbit, and includes the draft design of relevant sub-systems. Three optimizing loops follow for estimation of system budget, definition of packaging and relocation of routing (Fig. 10.11). Due to the multidisciplinary nature of satellite

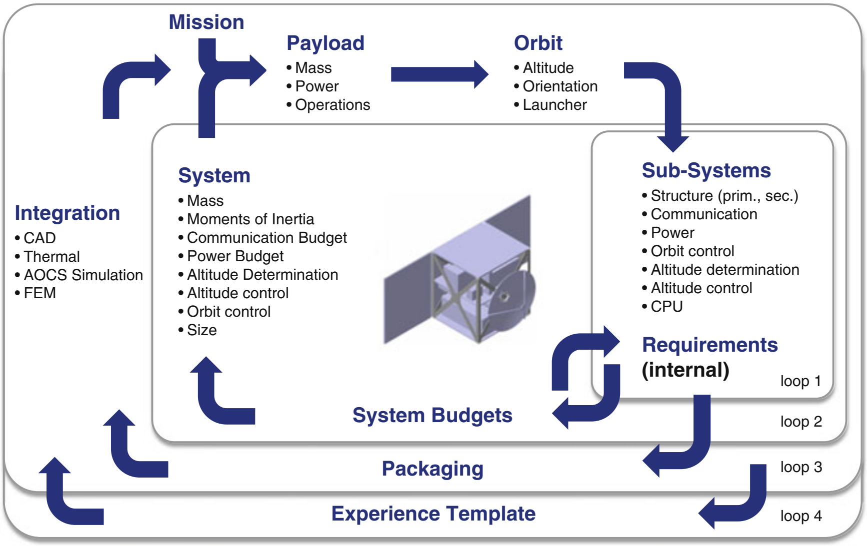


Fig. 10.11 Extended satellite design cycle [35]

design and the mostly proprietary formats of engineering software tools, a unified central product data model, which captures the knowledge, could be beneficial if it easily allows derivation of domain specific sub-models. A semantically rich and more abstract central data model from which the application program data can be generated in a straightforward way is adopted using the graph-based design languages in UML for complex system design [35]. The engineering objects represent the vocabulary, defined in UML, and the required model transformations represent the rules, i.e. the grammar of the design language.

From this high-level central data model, different interfaces derive the simulation models, as demonstrated in the field of geometry (CATIA/OpenCascade), thermal simulation (Esatan-TMS) and behavioral analysis (Matlab-Simulink). The compilation of the design language is done with the "DesignCompiler 43", an eclipse-based software tool for the compilation of design languages formulated in UML. This compiler processes design languages on all abstraction levels to the full detail of the simulation models. This yields a completely automated process within a unified framework. For the implementation of the design language a strong engineer's skill is necessary, while the generation of the models is routine compiler work. Thus, more abstract design decisions (e.g., topological ones) can be automatically compiled into their implementations in the different domains.

With the method of graph-based design languages in UML, the upfront investment is much higher compared to the application development based on standard KBE templates in CATIA. This lies in high efforts for the definition of the grammar. However, the generation of alternative design variants becomes very straightforward. It is possible to change some initial parameters, as for example the size of the satellite structure, while all subsequent design processes remain fully executable.

While in the approach of graph-based design languages, UML models are used and transformed, the performance of operations on the UML models are crucial for the performance of the design compilation process. In the present case, the models of the satellite are not larger than a few megabytes, even with the high complexity including a detailed geometry model. The largest model is the model for the generation of orbit simulation with 17 MB. This model is still performing well even though the used open source UML editors are not specifically designed and optimized for large data amounts. When complex geometry objects are created, they can for example be saved as a STEP file and integrated as an existing component. With this technique of model transformations and the adequate level of abstraction in information representation it is even possible to develop other models with similar complexity.

10.6.4 Shipbuilding

The design and manufacture of perfectly curved shell plates, which constitute a ship's bow and stern, is essential for low resistance. In the past these plates with extremely arbitrary 3D shapes were produced by plastically deforming through heating and water-cooling based on the wooden template check by human eyes. This carried great risks for the subsequent heat-sealing process. Based on the quantitatively immeasurable check-with-eyes results, how to decide the heating and cooling areas of the following processing is still uncertain. The variation in after-processing shapes arises and may cause gaps between the curved shell plates for the following heating sealing process, and the construction schedule's extension.

A software system for automated generation of curved shell plate's processing plan using virtual templates and laser scanner (Fig. 10.12) generates design data with the same shape as the real wooden templates, extracts a curved shell plate from 3D point cloud data measured by laser scanner [36, 37]. The generated virtual templates are located into the extracted point cloud, and finally two views (the curvature evaluation view and torsion evaluation view), which help the workers to decide the heating area for the following processing, are provided. Based on the torsion and curvature evaluation views, workers can decide the subsequent processing plan. This loop repeats until the curved shell plate fits the design shape.

10.6.5 Electrical Engineering

Electrical harness routing is a complex and largely manual task with numerous governing rules, while best practices for wiring looms typically comprise hundreds

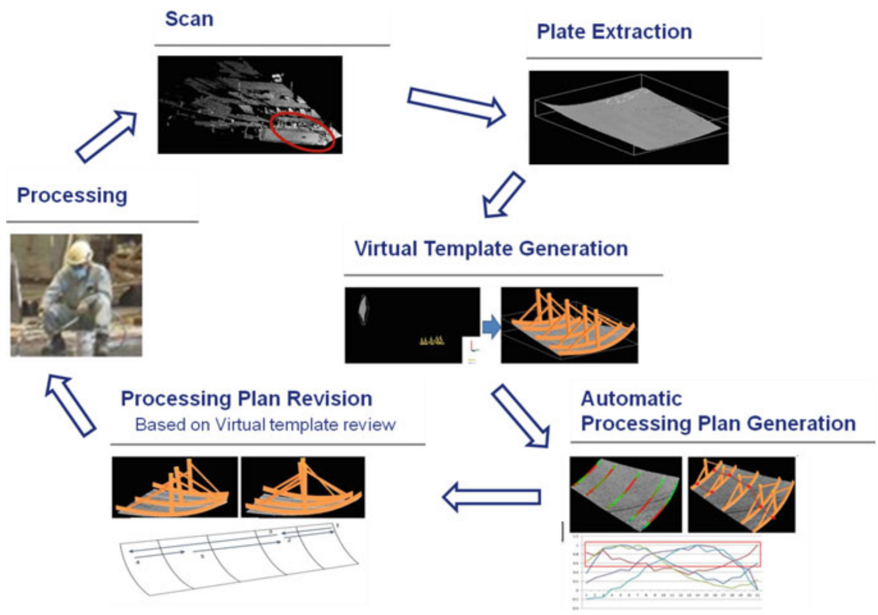


Fig. 10.12 Manufacturing processing flow with virtual template [36]

of harnesses all of which are manually routed. The generic process for harness routing involves manually creating a set of points in the CAD structural model from which the harness will be clamped to the main structure. The spine of the harness is passed through these points; ensuring sufficient clearance is given from the structure, particular subsystems, certain types of harnesses, moving parts, and areas of high heat. The process can be largely trial-and-error and often the only way to determine whether sufficient clearance has been allowed, is to make manual measurements in the CAD model which can be time consuming. These characteristics make the routing task a prime candidate for process automation.

The tool developed for the defense domain (Fig. 10.13) supports path routing from numerous domains including electrical harnesses, hydraulic and pneumatic pipes, and fuel lines and incorporates path-finding techniques from microprocessor routing and game AI domains, together with knowledge modeling techniques for capturing design rules [38]. The use case consists of an internal weapon storage area, which is complicated and densely populated with assemblies consisting of complex metallic structures, the payload envelope, and various subsystems and equipment with hundreds of interconnecting electrical harnesses and pipes.

Geometry is represented internally by the tool as a discrete, grid-based maze object with each grid node characterised by an integer address and node type. Categorisation of nodes within the solver is significant as it allows various obstacle types to be treated independently by rules implemented in the path-finding algorithm, such as following certain types and avoiding others.

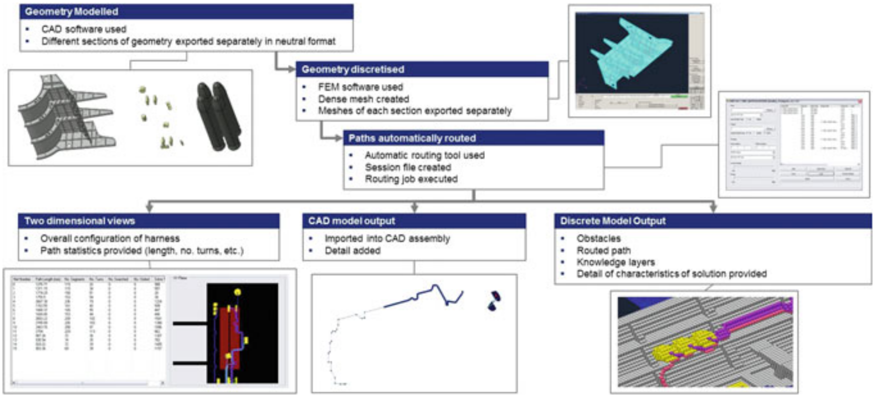


Fig. 10.13 Automated router system structure [38]

The rule editor for modeling domain rules consists of a simple form containing a set of controls for defining rule types, conditions for validity, area of influence, and action to be taken. Families of rules for different routing domains and path types are stored in separate libraries. The routing algorithm extends the popular A* search algorithm used in shortest path problems in game AI.

The system delivers output paths in three ways:

- A simple three-view path diagram referenced against input geometry is given in the software application itself.
- Resultant path spines are output as CAD-readable wireframe IGES models consisting of straight segments connected with a user defined bend radius.
- A finite element (FE) model comprising several component layers including input obstacles, routed paths and knowledge components describing the automated routing process can be used for further processing.

After a number of test runs of the system with various rule combinations were conducted, it was found that quality of resultant paths is closely coupled with the weight factor applied to individual rules.

10.6.6 Dentistry

The surgical process of dental implants is a delicate and accurate engineering task that can be automated in the early steps before manufacturing [39]. The challenge of the creation of the biomedical model is the complexity of the geometrical modelling dealing with natural shapes (see Chap. 25). A highly automated process to build an anatomical skull prosthesis piece in the CAD systems for manufacturing is developed based on bone’s border shape in a Computed Tomography (CT) slice (see Chap. 12). The arc that fills the correspondent failure in the bone border is extracted from the respective adjusted ellipse to each CT slice and the set of those extracted arcs can be superimposed to define the stack of images to build a 3D CAD model. Evolutionary algorithms were also applied to improve the quality of the data generated. A prototype was implemented by an open source Java-based tool (ImageJ) in order to create synthetic defects to simulate problems in the 3D virtual skull model. In the context of product development this approach brings an essential integration between design and manufacturing processes to reduce the elapse time among the medical procedure, modelling and machining. Elapse time among the medical procedure, modelling and machining.

10.6.7 Manufacturing

The Knowledge Optimized Manufacture And Design (KNOMAD) is a methodology for the analytical utilisation of manufacturing knowledge within design [40].

Such a manufacturing KM solution should at least fulfill the following major objectives:

- Transfer manufacturing knowledge to the upstream design disciplines
- Provide the knowledge base and rationale(s) from which manufacturing rules/ constraints/algorithms for use in engineering applications can be extracted
- Enable early estimation and optimization of design based on manufacturing parameters and knowledge.

The case study was carried out for a moveable control surface used on an aircraft, e.g., a flap, aileron, rudder etc. This use case was then developed to perform a trade-off study between two fastening processes that could be utilised in a manufacturing design solution: traditional riveting and friction stir welding (FSW). The actual analysis process taken in implementing manufacturing KM within a facilitating KBE framework is shown in Fig. 10.14. The main analysis output is manufacturing cost. Furthermore, geometry is assessed for interfaces that are to be joined or fastened together, along with any general manufacturing knowledge and data that needs to be used. The product is structured in the knowledge base according to the product architecture, so that part geometry and associated manufacturing knowledge is made available for the global product model as well as for the particular localized analytical model. The cost estimation process then investigates each of the interfaces by inputting the local geometry to cost estimating functions that are stored in libraries that can be viewed as specific process knowledge repositories. As stated, the analysis was run for two fastening processes, traditional riveting and FSW.

The results demonstrate that FSW is estimated to be almost three times faster and 20 % cheaper. That result of both production rate and cost improvement would probably not be captured at the same time (concurrently) by industry, or at least not on a repeatable basis.

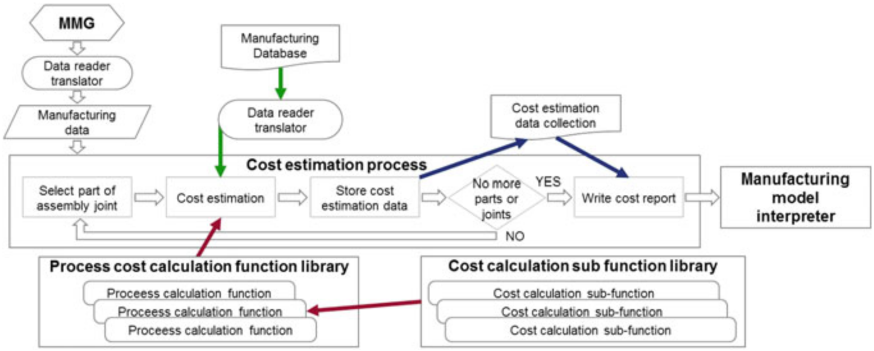


Fig. 10.14 Modelling process for the KBE assessment of manufacturing costs [40]

10.7 Discussion and Future Perspectives

Despite of its huge achievements described in the use cases above, KBE is still subject to intensive research and practical improvements. The valuation of KBE can be facilitated in two directions: KBE as autonomous procedure (application) and KBE as component of the business process as a whole.

When considering KBE autonomously, a number of developments have put KBE on a stronger footing over the last decade. In particular, the development and uptake of KBE-dedicated languages, the uptake of KBE language or template functionality in most major CAD systems and the development of supporting methodologies such as MOKA have strongly contributed to the uptake of KBE [5]. However, a number of challenges remain to be solved.

First of all, the justification of KBE application development, use and maintenance is currently not supported by scientific approaches [9], though recently advances have been made to support the identification of knowledge change characteristics [7] and to offer methodological support [41, 42].

Second, current KBE applications or templates (within CAD systems) only weakly address the issue of workflow integration. The European iPROD project [12] aims to address this through the development of models and tools for workflow integration into KBE. The project iPROD also considers KBE as a service (Software-as-a-Service, SaaS), thereby addressing a third challenge in KBE research, which is to ‘support web collaborative solutions and open source initiatives’ [5].

A final research challenge is to address the ‘black-box’ phenomenon of KBE development and use [9]: supporting the integration of code and documentation generation [5] as well as explicitly linking knowledge base structure and meaningful content with the KBE application elements and code.

When considering KBE as a component of a business process, many issues have arisen in the past and still remain unresolved. As KBE uses either additional CAD entities or additional software, which must be installed (Sect. 10.5), in the case of a PDM-controlled development process, KBE systems must be subjected to the PDM system. That is a serious issue because current PDM systems do not sufficiently support the KBE workflow [26]. Therefore a certain self-limitation of the used KBE functionality is necessary to remain in conformance with PDM, at least until PDM systems are able to support the whole KBE functionality.

Another issue is the complexity of the KBE data model and the corresponding relationships between singular entities. Although KBE can be applied easily like a blackbox, misunderstandings and mistakes can arise if the user has to derive the CAD model created by instantiation of a KBE template using complex relationships he is not familiar with. For example, more than 2,500 links are needed to define an entire body-in-white structure within a concept template [26]. This link management gives the capability of dividing complex structures into template—based and usable part structures. This issue could be resolved by use of augmented reality (AR) techniques that enable the multimedia representation of knowledge, especially interactive animation of 3D CAD models [43]. AR provides the changing views of virtual data—especially 3D models—in a real environment and allows the user to better understand the presented virtual data and knowledge in a more comprehensive way.

A further breaking point in the comprehensive use of KBE lies in engineering collaboration (see Chap. 7). Commercial KBE applications are not prepared for the exchange of templates on a regular basis (e.g., within the supply chain). Therefore each partner in the supply chain is forced to act autonomously [44]. Furthermore, if KBE entities shall be exchanged between two applications (e.g., different CAD systems), it would fail because there is no standard interface for such an exchange. Apart of the first draft of KBE Services for PLM as created by the object management group (OMG) [45], there is no known standardization activity.

Finally the request for intellectual property protection is the main obstacle for wide implementation of KBE in many industries now (see Chap. 18). As enterprises fear to lose their intellectual property, they block access for potential attackers, before it is even opened for the first time.

10.8 Conclusions and Outlook

As the preceding sections and use cases indicate, the use of KBE applications and templates may achieve significant results through automation of complex engineering tasks, significantly decreasing engineering task time while retaining product and process knowledge. As can be observed from the use cases, knowledge from downstream disciplines such as manufacturing can be implemented into KBE. By speeding up the design process and implementing downstream disciplines' knowledge, KBE is an enabler of the 'Design for X' approach—and by extension, KBE is a practical approach for implementation of the Concurrent Engineering philosophy [46]. Recent developments like CATIA V6™ allow use of templates for each module in this package, theoretically allowing automation of each task. In such way KBE becomes ubiquitous.

Recent research into KBE has focused on the structure and maintainability of KBE applications, as well as the progress in integrating KBE functionality in commercial CAD systems through the template approach. The use cases referenced and presented in this chapter have (partially) incorporated these research aspects.

A number of research challenges have been highlighted in the previous section. These challenges present themselves along the KBE lifecycle identified in Sect. 10.2. First of all, the justification of KBE development requires a sounder scientific footing. Capturing, formalization and implementation (packaging and distribution) of knowledge into KBE applications will require greater attention to workflow modeling, code documentation, IP concerns and integration with other business applications in PDM/PLM platforms. Finally, when considering the introduction and use of KBE applications and templates, the adaptability and maintainability of developed applications and templates remains a point of concern. However, it is anticipated that current EU-funded research such as the iPROD project as well as the mounting volume of KBE application development and associated academic output will lead to progress on these research challenges.

References

1. Hicks BJ, Culley SJ, Allen RD, Mullineux G (2002) A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design. Int J Inf Manage 22(4):263–280
2. Alavi M, Leidner DE (2001) Review: knowledge management and knowledge management systems: conceptual foundations and research issues. MIS Q 25(1):107–136
3. Birkinshaw J, Sheehan T (2002) Managing the knowledge life cycle. MIT Sloan Manage Rev 44(1):75–83
4. Liese H (2004) Wissensbasierte 3D-CAD Repräsentation. PhD thesis, Shaker Verlag, Aachen, TU Darmstadt
5. La Rocca G (2012) Knowledge based engineering: between AI and CAD. Review of a language based technology to support engineering design. Adv Eng Inform 26(2):159–179
6. Kuhn O, Liese H, Stjepandić J (2011) Methodology for knowledge-based engineering template update. In: Cavallucci D, Guio R, Cascini G (eds) Building innovation pipelines through computer-aided innovation. Springer, Heidelberg, pp 178–191
7. Verhagen WJC (2013) An ontology-based approach to knowledge lifecycle management within aircraft lifecycle phases. PhD dissertation, Delft University of Technology
8. Ouertani MZ, Baïna S, Gzara L, Morel G (2011) Traceability and management of dispersed product knowledge during design and manufacturing. CAD Comput Aided Des 43:546–562
9. Verhagen WJC, Bermell-Garcia P, Van Dijk REC, Curran R (2012) A critical review of knowledge-based engineering: an identification of research challenges. Adv Eng Inform 26(1):5–15
10. Skarka W (2007) Application of MOKA methodology in generative model creation using CATIA. Eng Appl Artif Intell 20:677–690
11. Stokes M (2001) Managing engineering knowledge—MOKA: methodology for knowledge based engineering applications. Professional Engineering Publishing Limited, London
12. iProd (2013) Improving the product development process (PDP—iPROD project). Retrieved 20 Mar 2013, from http://www.iprod-project.eu/index

13. Van Dijk R, Zhao X, Wang H, Van Dalen F (2012) Multidisciplinary design and optimization framework for aircraft box structures. In: Proceedings of conference on 3rd aircraft structural design, Delft
14. Bermell-Garcia P, Fan IS (2008) Practitioner requirements for integrated knowledge-based engineering in product lifecycle management. Int J Prod Lifecycle Manage 3:3–20
15. Vermeulen B (2007) Knowledge based method for solving complexity in design problems. PhD thesis, Delft University of Technology
16. Uschold M, Gruninger M (1996) Ontologies: principles, methods and applications. Knowl Eng Rev 11:93–136
17. Matsokis A, Kiritsis D (2010) An ontology-based approach for product lifecycle management. Comput Ind 61:787–797
18. Kitamura Y, Kashiwase M, Fuse M, Mizoguchi R (2004) Deployment of an ontological framework of functional design knowledge. Adv Eng Inform 18:115–127
19. O'Connor MJ, Nyulas C, Tu S, Buckeridge DL, Okhmatovskaia A, Musen MA (2009) Software-engineering challenges of building and deploying reusable problem solvers. Ai Edam-Artif Intell Eng Des Anal Manuf 23:339–356
20. Lee JH, Suh HW (2008) Ontology-based multi-layered knowledge framework for product lifecycle management. Concur Eng Res Appl 16:301–311
21. Curran R, Butterfield J, Jin Y, Collins R, Burke R (2010) Value-driven manufacture: digital lean manufacture. In: Blockley R, Shyy W (eds) Encyclopedia of aerospace engineering. Wiley, Hoboken
22. Ruschitzka M, Suchodolski A, Wróbel J (2010) Ontology-based approach in hybrid engineering knowledge representation for stamping die design. In: Pokojski J et al. (eds) New world situation: new directions in concurrent engineering, Proceedings of the 17th ISPE international conference on concurrent engineering, Springer, London, pp 205–212
23. Bermell-Garcia P, Verhagen WJC, Astwood S, Krishnamurthy K, Johnson JL, Ruiz D, Scott G, Curran R (2012) A framework for management of knowledge-based engineering applications as software services: enabling personalization and codification. Adv Eng Inform 26:219–230
24. Verhagen WJC, Bermell-Garcia P, Mariot P, Cotton JP, Ruiz D, Redon R, Curran R (2012) Knowledge-based cost modelling of composite wing structures. Int J Comput Integr Manuf 25:368–383
25. Liese H, Stjepandić J (2004) Konstruktionsmethodik: Wissensbasierende 3D-CAD-Modellierung, CAD/CAM Report. Dressler Verlag, Heidelberg, 10. http://www.prostep.com/de/prostep/medien/cad-cam_kbe.htm. Accessed 15 Feb 2014
26. Katzenbach A, Bergholz W, Rohlinger A (2007) Knowledge-based design—an integrated approach. In: Krause FL (ed) The future of product development. Proceedings of the 17th CIRP design conference, Springer, Heidelberg, pp 13–22
27. Poorkiany M, Johansson J, Elgh F (2013) Implementing engineering design automation: identified issues. In: Bil C et al. (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 323–332
28. Elgh F (2013) A task oriented approach to documentation and knowledge management of systems enabling design and manufacture of highly customized products. In: Bil C et al. (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 119–128
29. Biahmou A (2012) An efficient CAD methodology for glove box design. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering (2013). Springer, London, pp 219–229
30. Schemainda K (2003) Wissensbasiertes Parametrisieren eines Faustsattels, Diploma Thesis, Berufsakademie Mannheim
31. You CF, Jeng CR, Liu KY (2013) Design for automotive panels supported by an expert system. In: Coelho DA (ed) Advances in industrial design engineering. InTech, Rijeka, pp 187–210

32. Rohwäder T (2007) CAVA erleichtert die Einhaltung der gesetzlichen Vorschriften, CAD-CAM Report Mai 2007, pp 20–26
33. Brüning HC, Liese H (2013) Reliable Methods for the virtual car design process in the conceptual development of passenger cars at volkswagen AG. ProSTEP iVip Symposium, Hannover, 16–17 April. http://www.prostep.org/fileadmin/user_upload/ProSTEPiViP/Events/Symposium_2013/Presentations_ProSTEP-iViP-Symposium-2013.zip. Accessed 15 Feb 2014
34. Ugarte D, Izaguirre A (2013) Multistate feature modelling of a very complex design feature. In: Abramovici M, Stark R (eds) Smart product engineering. Springer, Heidelberg, pp 451–461
35. Groß J, Rudolph S (2012) Dependency analysis in complex system design using the FireSat example. In: Proceedings of the 22nd annual INCOSE international symposium (IS 2012), 9–12 Jul. http://www.isd.uni-stuttgart.de/forschung/publikationen/publi/2012_06_11_INCOSE-DependencyAnalysis.pdf. Accessed 15 Dec 2013
36. Sun J, Hiekata K, Yamato H, Nakagaki N, Sugawara A (2013) Automatic generation of curved shell plates' processing plan using virtual templates for knowledge extraction. In: Bil C et al. (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 441–450
37. Sun J, Hiekata K, Yamato H, Nakagaki N, Sugawara A (2014) Virtualization and automation of curved shell plates manufacturing plan design process for knowledge elicitation. Int J Agile Syst Manage 7(3/4):282–303
38. Van der Velden C, Bil C, Yu X (2011) A knowledge-based approach to design automation of wire and pipe routing through complex aerospace structures. In: Frey DD et al. (eds) Improving complex systems today. Proceedings of the 18th ISPE international conference on concurrent engineering, Springer, London, pp 225–232
39. Szejka AL, Rudek M, Canciglieri Jr O (2012) A reasoning system to support the dental implant planning process. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering (2013), Springer, London, pp 909–919
40. Curran R, Verhagen WJC, Van Tooren MJL, Van der Laan TH (2010) A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD. Expert Syst Appl 37:7336–7350
41. Johansson J, Elgh F (2013) How to successfully implement automated engineering design systems: reviewing four case studies. In: Bil C et al. (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 173–182
42. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manage 7(3/4):324–347
43. Januszka M, Moczulski W (2012) Acquisition and knowledge representation in the product development process with the use of augmented reality. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering (2013), Springer, London, pp 219–229
44. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manage 7(2):116–131
45. Bermell-Garcia P (2007) A metamodel to annotate knowledge based engineering codes as enterprise knowledge resources. PhD thesis, Cranfield University
46. Van Der Velden C, Bil C, Xu X (2012) Adaptable methodology for automation application development. Adv Eng Inform 26(2):231–250

Chapter 11
Product Lifecycle Visualization

Alfred Katzenbach, Sebastian Handschuh, Rudolf Dotzauer and Arnulf Fröhlich

Abstract When products are developed in 3D using engineering applications, the data is initially stored in the native format of the used CAD software. If this 3D CAD data is to be shared with people who do not have this software or to be consolidated with visualization data from other sources, neutral 3D formats are needed. For visualization of product data in the engineering field—regardless of native CAD formats—a plethora of 3D formats are available. Among these are disclosed or standardized formats like PDF from Adobe, JT and also X3D, Collada and STEP. The choice of a format has many implications, including the options available for using the data and the resulting follow-up costs. For this reason product lifecycle visualization is a rising discipline for product lifecycle management. This chapter provides an overview of the industrial challenge, technical background and standardization, typical applications, and evaluation and testing in the field of engineering visualization with neutral 3D formats. The chapter is completed by assessment approach for 3D formats and examples from the industrial practice in various fields.

Keywords Engineering visualization · Interoperability · Data exchange · JT standard · Supply chain · Engineering collaboration

A. Katzenbach (✉)
Katzenbach Executive Consulting, Gaienhofen Germany, University Stuttgart, Gaienhofen, Germany
e-mail: alfred@katzenbach-web.com

S. Handschuh
Daimler AG, Hanns-Klemm-Str. 5, 71034 Böblingen, Germany
e-mail: sebastian.handschuh@daimler.com

R. Dotzauer
Continental, Ringlerstr. 17, 85057 Ingolstadt, Germany
e-mail: rudolf.dotzauer@continental-corporation.com

A. Fröhlich
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: arnulf.froehlich@prostep.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_11

11.1 Introduction

Product lifecycle visualization (PLV) evolved over the years from the field of Computer Graphics—in particular 3D Computer Graphics—which is a part of computer science. Computer Graphics produces visual evocations of virtual worlds that look real by creation and processing of arbitrary data structures to specific visual presentations (2D or 3D) using computer devices. William Fetter was credited with coining the term computer graphics in 1960 to describe his work at Boeing [1].

The field of Computer Graphics has evolved continuously over the past five decades following the development of a huge collection of algorithms, methods, and techniques for various applications in modeling, animation, visualization, face recognition, real-time rendering and game engine design [2]. Demands from various industries (ICT, consumer electronics, movie, games, automotive, aerospace etc.) have continuously fuelled this growth. Advances in processor technology and graphics hardware have enforced this tendency making 3D computer graphics ubiquitous.

3D computer graphics are—in simplified terms—often referred to as 3D models, but there are significant differences. A 3D model is the mathematical representation of any three-dimensional object which can, in addition, (e.g. in case of computer-aided design) encompass complex design rules (see Chap. 10). A model is not technically a graphic until it is displayed: graphic is presentation of the (product) representation. It emerges by translation (conversion) of a model. Due to 3D printing, 3D models are not confined to virtual space. A model can be displayed visually as a two-dimensional image through 3D rendering, or processed in non-graphical computer operations (simulations and calculations) [3].

The development of interactive 3D CAD systems is deeply embodied in achievements with 3D computer graphics. A powerful graphics engine for visualization, rendering, and animation has become an important basic module of each modern CAD system. In terms of CAD usage in concurrent engineering (CE), two basic activities can be distinguished: authoring (modeling, design) and consumption (access, browsing, visualization, animation). For a long time, both activities ran in parallel on the same workstation serving other users with plots and hard-copies. With falling prices for hardware, 3D CAD could become ubiquitous and many companies faced the challenge to finally replace the drawings by appropriate 3D models. Analysis of industry practices involving product development and delivery has disclosed that the count of 3D consumers is more than 10 times larger than the count of 3D creators [4] (Fig. 11.1). However, typical consumers of 3D data were inhibited by the operational complexity of modern CAD systems, high efforts for training and long response times on large product models. Thus, in the early 2000s, the question arose: “Does everyone need a full CAD system?” Although the answer was quite simple, no action occurred for a while because of the huge variety of 3D formats and applications which are focused on special aspects and purposes rather than on process chains. Thus, the challenge arises to accompany the CAD system

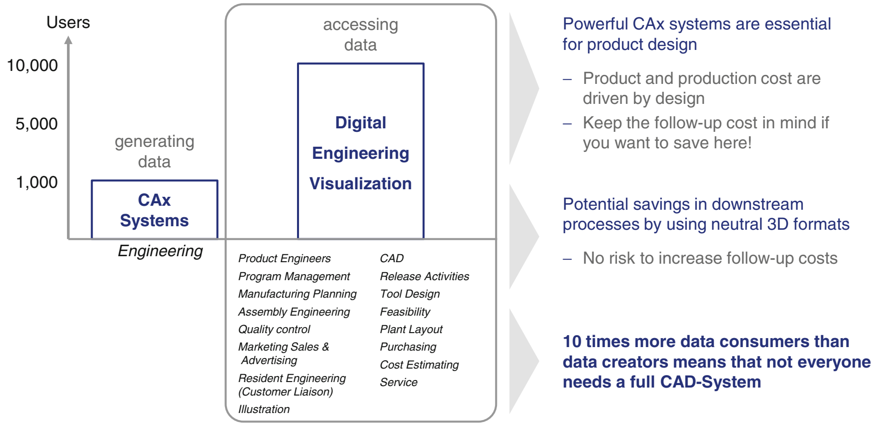


Fig. 11.1 Application field of 3D technologies in engineering

and its native format with a standardized 3D process format which serves the downstream processes [Digital Mock-Up (DMU), engineering collaboration, process planning, CAQ etc.] in a uniform way. Partial attention is given in the later stages of product lifecycle (maintenance, repair, overhaul), as well as non-technical departments (purchase, marketing, aftersales) which need comprehensive visual information.

Reflecting these developments, this contribution has two aims. First, research, development and standardization challenges relative to PLV from a lifecycle perspective are identified. Second, recent practical applications of PLV addressing these challenges in various industries are highlighted. In particular, the issues of definition of a downstream process, fulfilling the needs of plethora of manufacturing-related applications and integrating PLV in engineering collaboration are discussed.

The structure of this chapter reflects these aims. In Sect. 11.2, Engineering Visualization and its context and challenges are briefly introduced. Technical background and advances in international standardization are discussed in Sect. 11.3. Subsequently, application of 3D formats in engineering domains is explained in Sect. 11.4. Evaluation and testing of 3D formats for engineering visualization and related institutions are highlighted in Sect. 11.5. Section 11.6 showcases the results of assessment of 3D formats for various industries. Industrial use cases which give broad insight in practical applications follow in Sect. 11.7. A discussion chapter gives insight into future directives where such formats can be used. Finally, an outlook is given with respect to the future of product lifecycle visualization from a CE perspective.

11.2 Engineering Visualization: Context and Challenges

In the last decade the technology of CAD translation has reached a high level of maturity and robustness. However there is still a vigorous demand for further development, especially in the use of CAD data in the downstream process. The usage of DMU as an important and well established validation process has unfolded further potential for improvement (see Chap. 13). Nowadays CAD models are already characterized by an easy use even with their high level of complexity. In the past, the high level of complexity combined with a necessity for easy usage has offered several 3D visualization formats to enter the market. Therefore, Siemens PLM has adopted JT, Dassault Systèmes 3DXML, PTC Productvision, Adobe PRC as 3D component to well-known PDF (3D PDF), Autodesk DWF, and Lattice has adopted XVL. In addition, the far-reaching use of 3D formats by efficient translators and suitable workflows was simplified by well-known PDM systems (e.g. Siemens PLM Teamcenter, PTC Windchill, Dassault Systems Enovia, SAP PLM, etc.). Hence, these 3D formats can facilitate several process chains ranging from designing to product simulation, validation, manufacturing and downstream life cycle stages.

The old discussion, whether a unique format can meet the requirements of engineering cooperation (internal and external), was sparked by the large-scale penetration of visualization formats in the market. An attempt is presented by 3D digital engineering visualization (3D DEV). It enables all users taking part in the product development and distribution process to access the 3D models without logging into the convoluted computer aided design (CAD) authoring systems. STEP has gathered a broad support and high stake for the CAD translation, however, due to its technological limitations it is not suitable for viewing purposes. Lastly, users must be consulted with which formats are the most suitable in their particular case [5, 6]. There are typical use cases for a functional evaluation of these formats:

- Viewing of engineering data
- Design in context
- Data exchange between partners in the supply chain
- Packaging and digital mock-up (DMU)
- Documentation and archiving
- Use in the portable PLM document (i.e. use of 3D and additional information in domains related to engineering; see Fig. 11.2).

In cases where administrative and geometrical information be obtained certain problems arise, e.g. engineering change among partners in the supply chain. These discrepancies lead to a difficult coordination of release and change process between collaborating companies. One issue involves the proposition of changes, comprehension of those changes, and transmission of change orders. Another problem lies within the exchange of related data from distinctive authoring system, data bases, and enterprise management systems. A third issue is the implementation of

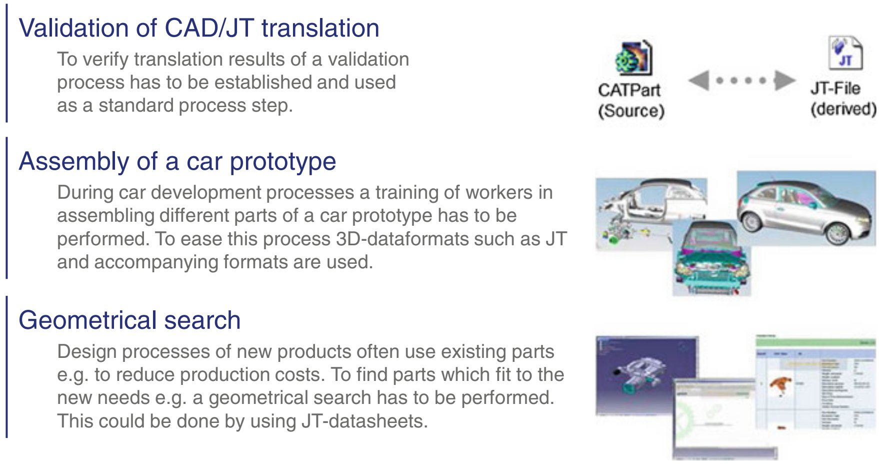


Fig. 11.2 Typical use cases of JT [13]

approved changes within the 3D model geometry and related part lists, work instructions, materials property sheets, notations as well as other product manufacturing information (PMI) source data in assorted native file formats in different information systems.

In the last decade the JT file format, originally developed by Siemens PLM Software, evolved to a de facto standard for 3D DEV for the automotive and aerospace sector. In 2010, the Global Automotive Advisory Group (GAAG), a forum of international managers of automotive engineering IT, identified the industry endorsement of JT and urged Siemens PLM Software to disclose the JT file format definition to the International Standards Organization (ISO) for recognition as an international standard for 3D DEV. To drive the ISO acceptance of the JT format forward, an internationally accepted leader in the development of engineering norms and standardization of engineering collaboration processes, the ProSTEP iViP Association (PSI), took operational lead and supervised the project. Through international support, the project was successfully concluded in December 2012 with the acceptance of JT as ISO IS14306:2012. This acceptance led to a growing user interest in quality and robustness, especially the interfaces. Therefore the subsequent explanation of standardization processes will concentrate on the JT format, even though alternatives such as the STEP, 3D PDF, and 3DXML are currently in different standardization phases.

The benefits of open standards are a reduction in total costs of ownership as well as the autonomy from provider and competition. When visualization data exchange is regarded, the use of JT and STEP AP 242 (ISO 10303-242) as complementing standards for lightweight visualization format for 3D industrial data as well as for product structure, meta, and kinematic data is strongly supported [5].

Another application is the well-known, already standardized PDF format which can be nowadays found on almost every PC. The official release of PDF is dated back to July 1st, 2000 and was published as ISO 32000-1:2008. Adobe Systems Inc. made the decision to broaden the scope of the PDF format through implementing 3D representation functions into PDF format (3D PDF) [7]. With this step they created smart PDF templates that unite the advantages of smart documents and 3D representation. This technology serves as an automation and support for proposals for change, change orders, and change notifications. Now, all the relevant data for analyzing and implementing change orders can be lodged in a single PDF container. Furthermore, 3D models, 2D drawings, parts lists, and all other types of information can be extracted and automatically read into the corresponding backend information systems. As a consequence, automation incorporates the cross-company change and release processes seamlessly into the domestic engineering change management systems at each of the partner's locations.

11.3 Technical Background and Standardization

Based on their industrial relevance, further explanation will be focused on the formats JT and 3D PDF.

11.3.1 Development and Technical Attributes of JT

JT with its binary character has a container structure whose data model facilitates several CAD geometry representations (Fig. 11.3). The JT user can store the different representations in a JT file together or separately [5, 6].

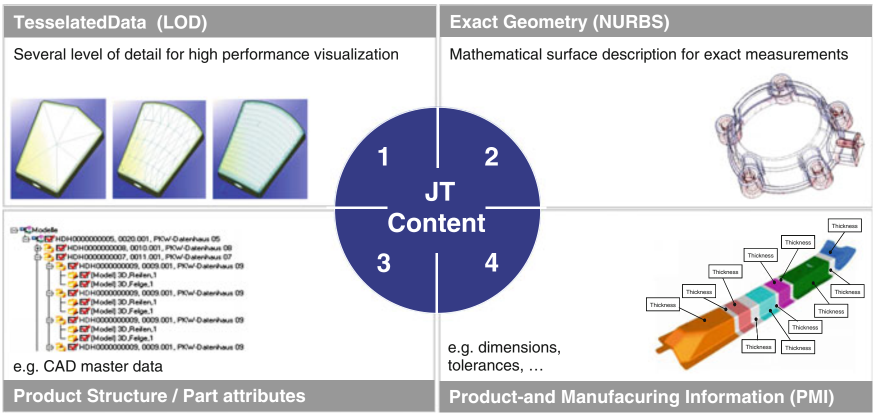


Fig. 11.3 Content of ISO 14306 (JT) [10]

- **Boundary Representation (BREP)**: BREP provides the highest level of representation precision. Data generated with BREP is compressed by using different algorithms and stored without loss. In its present specification (9.5), only two BREP types are allowed: the conventional JT-BREP representation and XT-BRED based on the Parasolid boundary representation. The XT-BREP will be favored in the further implementation of JT based software.
- **Tessellated Geometry**: Tessellated Geometry describes a faceted representation of surfaces and solids. Within a JT file different level of details (LOD) can be defined. JT files with a low LOD provide a lower representation precision, on the one side, however, have less volume of data, on the other side. From a very high LOD almost exact geometry can be concluded but with a large volume of data.
- **Ultra-Lightweight Precise (ULP)**: The newest compression method is represented by ULP. With ULP, the user takes the middle road between high level of representation precision and low volume of data. In comparison to tessellated geometry, the precision of 3D geometry is significantly higher whereas at the same time, the file size can be significantly reduced. The aim is delivering high quality surface geometry with only minor alterations of the original BREP geometry.

JT has one property of high importance: PMI consisting of 3D dimensions and annotations can be stored in JT files. This property, also an eminent feature of modern CAD systems, enables JT to archive all data that are required to substitute paper drawings. Hence, the growing use of JT could lead to the redundancy of drawings, either paper of CAD/raster, and create a fully 3D process chain.

Initially, the ISO has published the JT version 8.1 as a publicly accessible specification (ISO PAS 14306). The end of the ISO standardization process was the approval in December 2012 of the ISO standard (ISO 14306) for newly published JT version 9.5. It has been extended to include ULP specification and semantic PMI (product meta data) among other things.

As a result of creating data with a neutral 3D format, a considerable reduction in size can be observed. Altogether 15 test assemblies originated from distinct CAD systems were used to create different 3D formats [8]. The results indicate that the data volume is primarily defined by the content rather than the format itself.

There's no significant difference in volume of data, whether exact BREP data is converted to JT or 3D PDF. Likewise, when tessellated data is converted, the volume of both formats is roughly the same as after 3D XML conversion. The conversion of simplified BREP data to JT and 3D PDF yields the same data volume. Only in the case of conversion to STEP a significant reduction of the data volume through an external compression algorithm can be achieved. As a result, a higher precision leads to a larger file size as a universal rule.

11.3.2 Development and Technical Attributes of 3D PDF

3D PDF is the name of the PDF format which also provides native support for 3D data. In 2008, 3D PDF was published as the international standard ISO 24517-1 [8]. The following CAD geometry representations can be stored within 3D PDF:

- **Universal 3D (U3D)** was developed by Ecma International and is a compressed file format for 3D data that is natively supported by the PDF format. 3D objects in U3D format can be inserted into PDF documents and visualized interactively. The PDF standard supports the first and third editions of U3D; both these versions can include only tessellated geometry and animation data.
- **Project Reviewer Compressed (PRC)** is used to store representations as tessellated or precise (BREP) geometry. When converting to PRC, different levels of compression can be used. Support is also provided for PMI data including geometric dimensioning and tolerancing.

In PDF files, PRC is the preferred 3D format for implementing requirements in manufacturing industry. PRC ISO successfully completed the Draft International Standard (DIS) ballot process in December 2012. In addition to the JT format, 3D PDF offers all the options that conventional PDF offers (Fig. 11.4).

In addition to pure geometry representation the following could also be stored in 3D PDF:

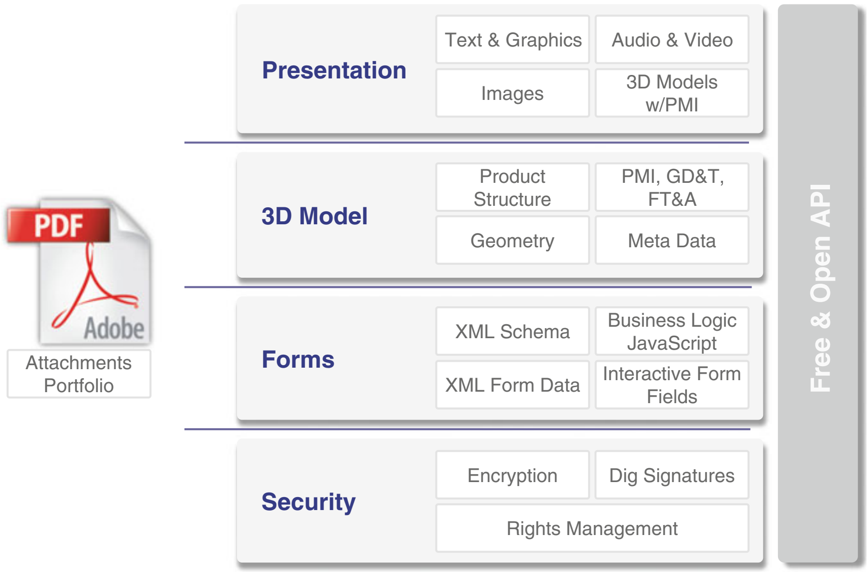


Fig. 11.4 3D PDF architecture [7]

- Multimedia data, e.g. video and audio files
- Access protection for document security, e.g. password protection
- Interactive forms, e.g. for collecting feedback or storing input
- Collection of arbitrary documents, e.g. Office or native CAD files.

This makes 3D PDF a format that extends well beyond the realm of engineering. Because 3D PDF allows 3D information to be represented together with other information, it offers many application possibilities. 3D data and PLM data are combined in a single document and can be displayed or even enhanced by the user.

11.4 Applications of 3D Formats

A sole provision of an international standard without any instructions on how to use it will have little value. Therefore, a process definition with several concrete use cases is fundamental for successful implementation of such an initiative. In this context, an extensive research of the downstream processes has been conducted to find the areas where the use of JT rather than native CAD data is of potential interest [9, 10].

A summary of potential work share between JT and native CAD is given in Fig. 11.5 within the scope of an international automotive group. A distinction has to be made between the integration processes among the OEMs and several suppliers and the value adding processes within the company, that still need native CAD representation. For those integration processes, JT is an alternative that can lead to a higher level of flexibility, faster process and cost cuts in particular.

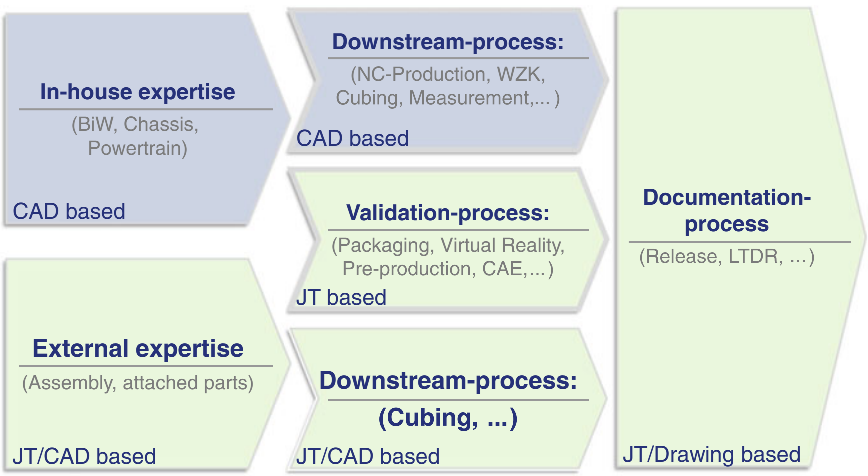


Fig. 11.5 Planned work share between native CAD and JT [10]

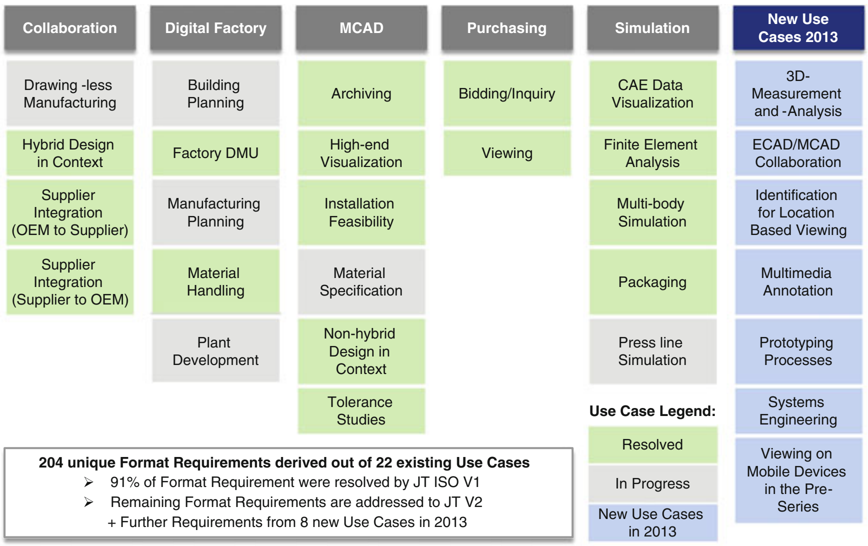


Fig. 11.6 Use cases for lightweight formats [11]

Additional studies were conducted that involved groups with several automotive manufactures as well as their suppliers. An analysis of the group's 22 use cases shows a notably high preference for four use cases that are discussed below in more details with examples [11] (Fig. 11.6).

Viewing: If usage of CAD systems is not desired, the visualization of engineering data with 3D viewers gains importance in different areas: viewing of single parts and assemblies, usage of 3D models to provide information of general kind, e.g. for bidding/inquiry and high-end visualization in virtual reality scenarios.

Depending on the specific application context the use case can vary. In the most cases a simple observation of the geometry is sufficient, where as in some cases the metadata or performance-intensive observations of large assemblies are more dominant. The list of most important requirements includes:

- Fulfillment of quality criteria for geometry precision.
- Availability of different level of details.
- Availability of detailed color information for whole component and single geometrical items.
- Storage of metadata like attributes.

Packaging/DMU: Within the DMU, the spatial product properties are analyzed and reviewed. This process can include the revision of the global geometry with respect to shape, dimensions, inference checks, collision checks for assembly/dis-assembly, and design space checks (Fig. 11.7).

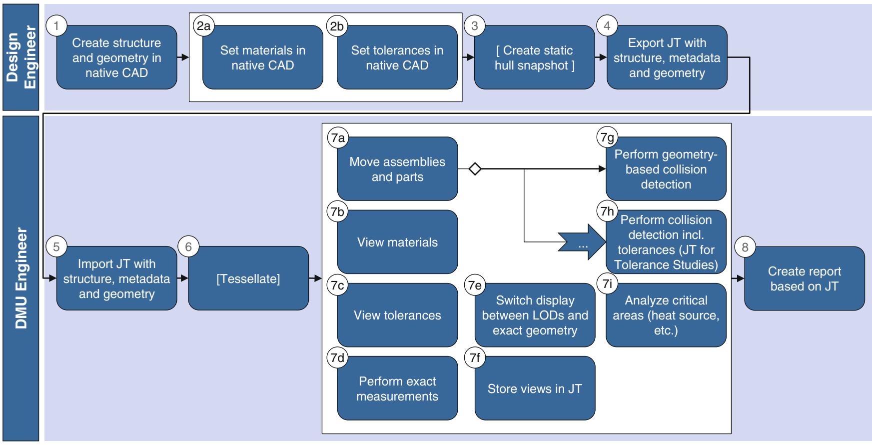


Fig. 11.7 Use case diagram "Packaging"

The representation of the product structure, geometry and metadata as well as the analysis is realized in a DMU application (see Chap. 13). The analytical results are summarized and stored in a separate report. The list of most important requirements includes:

- Utilization of models stemming from distinct source systems (multi-CAD)
- High quality examination of large assemblies
- Transmissibility of kinematics from source model to target model for dynamic DMU examination.

Design in Context: The design in the context of the existing geometry is a frequently used modeling method. Here, current CAD models (e.g. from previous design or from partners) are read into the CAD system. The design engineer creates a reference of the new or existing JT-geometry to the part or assemblies read in. References from native CAD to JT are also considered. These references can include linkages between auxiliary geometry (point, axis or plane), geometrical elements (edges, vertices or faces) and exact geometrical links (curves). In the following downstream processes the use of this assembled model as a consolidated individual unit is required. Therefore, it must be feasible to create technical drawings. The list of most important requirements includes:

- Storage of tessellated and exact BREP data
- Possibility of links between JT and native data
- Derivation of technical drawings from hybrid models.

Partner Integration: There is a subdivision of the use case into two sections. On the one hand, the goal of the "Supplier to OEM" use case presents the exchange of JT files to the OEM. On the other hand, the "OEM to Supplier" use case aims at the transfer verified JT geometry with all indispensable metadata to the supplier.

The two sections are similar in the way that they both provide a foundation for additional use cases. This implies that with the OEM or the Supplier more specific use cases will be employed afterwards. The list of the most important requirements includes:

- Verification with regard to the native CAD model and quality check of JT
- Inclusion of the meta data into the JT model
- Exclusion of intellectual property, when needed.

11.5 Evaluation and Testing

A crucial issue in stimulating the application and development of JT is a collaborative approach between the different bodies on the one side and a collaborative approach of the activities themselves on the other side.

11.5.1 Involved Bodies

The foundation for the JT standardization efforts was the tremendous commitment of international parties. Once the impetus by SASIG was provided [1], the four entities PSI, the German Association of the Automotive Industry (VDA), the Automotive CIO's and the Global Automotive Advisory Group for PLM (GAAG) have become the key players in this process (Fig. 11.8). Hereby, PSI (www.prostep.org) has put enormous effort in the pursuit of these activities [11].

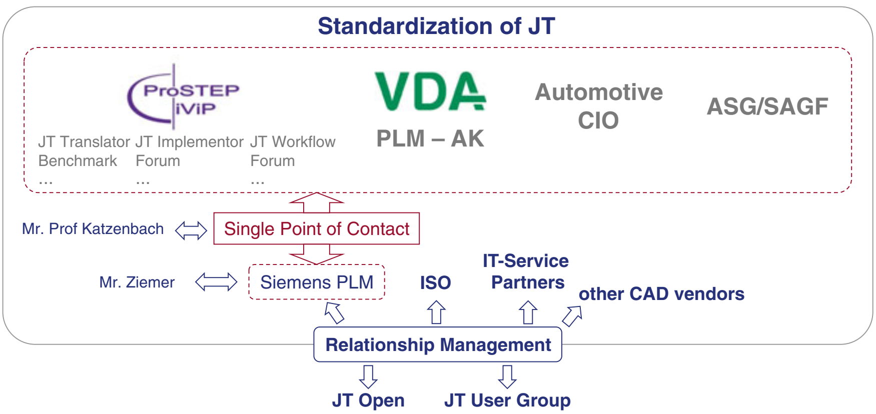


Fig. 11.8 International cooperation in the standardization of JT

Lastly, the Memorandum of Understanding (MoU) between PSI and Siemens PLM forms the basis between both entities in the area of JT. The aim of the memorandum is to advance the mutual understanding of goals and measures to make JT a mandatory format.

11.5.2 National and International Coordination

Back in 2009, a collaboration between the PSI and the VDA (German Automotive Association) resulted in the initiation of three different but interconnected worldwide unique JT activities. In the scope of the first activity, the JT workflow Forum, Use Cases are defined, representing the common view of the involved industrial partners. The second activity, the JT Implementor Forum, gives support for the development of the JT translator and administrates a platform for compatibility checking for the vendors. The third activity, the JT Application Benchmark, deals with the extended testing of the requirements within a well selected, repeatable, neutral environment [12].

Project schedules of these three activities are well-coordinated in order to advance the application of JT as a common effort.

JT Workflow Forum: The main goal of this project is the definition and prioritization of key use cases by identifying the important downstream processes for the data exchange based on JT data. Requirements are gathered and test criteria specified for all identified use cases, e.g. with regard to the exchange of visualization data, the validation of 3D geometry, Geometric Dimensions and Tolerances (GD&T) and the translator quality.

The Content Harmonization subgroup provides support for these tasks. The subgroup carries the responsibility for a precise definition of which requirements need to met by the JT format and by STEP AP242 as the backbone format and sets priorities for their implementation.

JT Implementor Forum: The maintenance of a platform for the vendors to check the compatibility is one of the main tasks of the JT Implementor Forum. On the platform, cooperating vendors are able to test their developed and highly confidential translators. The JT Implementor Forum as a neutral platform is aimed at software vendors, on which they can run benchmarks within an environment characterized by mutual trust. Further, they have the chance to share information on know-how they already have gained.

Additionally, the work effort put into the JT Implementor Forum by the vendors can be regarded as some kind of training for the JT Application Benchmark.

JT Application Benchmark: After the preceding standardization processes of JT as well as STEP AP242, the attention is shifting more to the assurance of the data exchange quality within a neutral Benchmark and support of the JT translator development.

A first JT translator benchmark (2009) was launched by the PSI in collaboration with the VDA PLM working group. In 2010, the second translator benchmark was performed and the third benchmark ended in 2012. The focus of the benchmark comprised three main topics:

1. Export of CAD to JT with focus on LOD and PMI
2. Viewing of JT with focus on performance and functionality
3. Converting assemblies with STEP AP242 XML and JT.

Cooperating vendors taking part in the benchmark are welcome to show their newest functionalities with Showcases in front of the members of the JT Workflow Forum [13]. At the moment, there is much attention paid to cutting-edge solutions which present the possibilities of the JT format and applications. Still, small functional errors are observed, which calls for improvement in the future (Fig. 11.9).

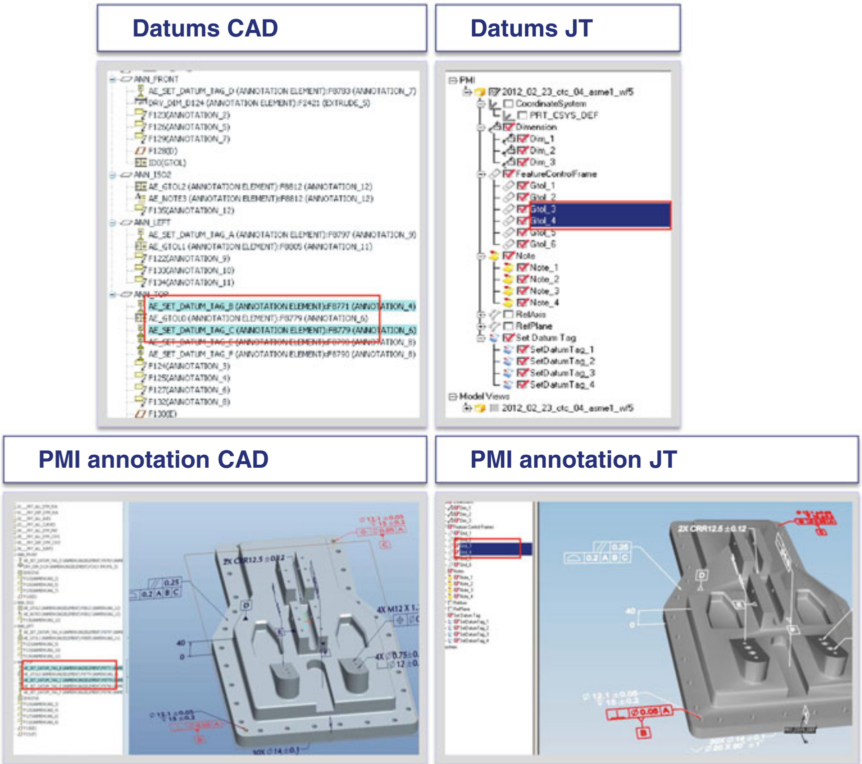


Fig. 11.9 Results of recent JT benchmark

11.6 Assessment of 3D Formats

A multitude of processes exists that benefit from neutral based data exchange of CAD data in engineering-related domains. Based on five typical use cases an analysis of common neutral 3D formats was done [8]. The four engineering formats STEP, 3D XML, JT and 3D PDF were reviewed and assessed for their suitability for the following application scenarios:

- Viewing
- Data Exchange
- Digital Mock-up
- Documentation & Archiving
- Portable PLM Document

The four formats were examined to determine the extent to which their properties are suitable for five of the most commonly occurring application scenarios in an enterprise. The usability for the scenarios was also the decisive factor to look at these four formats and to not take into account other formats like IGES, DXF and X3D that do not have strong distribution in the selected application scenarios.

After the rough evaluation of the technical specification of the visualization formats, an overall comparison can be drawn (Fig. 11.10).

As evaluation criteria, the functional requirements of the selected use cases were used. Additional criteria were the file size options for detailing, the collaboration opportunities in engineering and last but not least the current integration-level within PLM applications. A comparison of the 3D formats by singular properties is shown in Fig. 11.11.

Each of the neutral 3D formats has its strengths and thus offers advantages in one or more of the defined application scenarios. In particular, the expected areas of

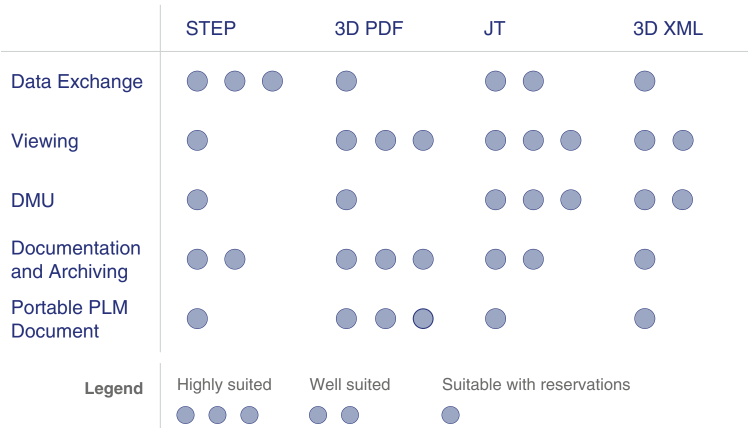


Fig. 11.10 Comparison of four visualization formats [8]

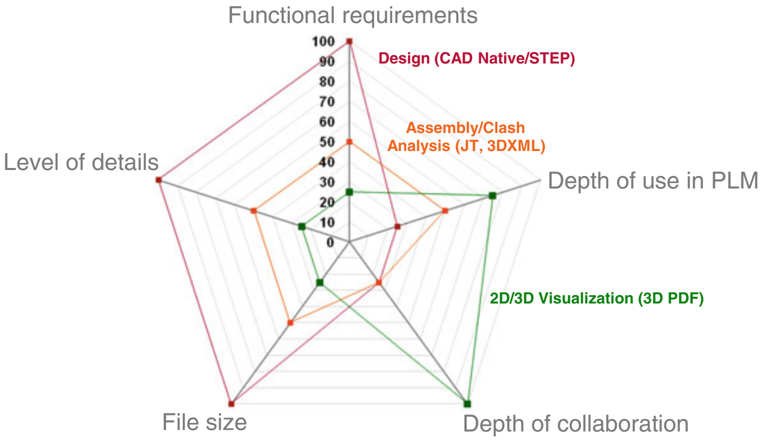


Fig. 11.11 Comparison of 3D formats by singular properties

application give an indication as to which format is the best choice. The weighting of the characteristics of a format will always be dependent on the overall context and the individual needs of a company. Depending on the scope, one 3D format or the combination of multiple 3D formats can be adopted.

11.7 Use Cases

3D lightweight formats have become accepted in many industries. Especially, in use cases of downstream processes (see Chap. 23) 3D lightweight formats offer massive benefits. In opposition to the core design process whereby CAD data are either created or edited, the subsequent phases of the product creation process contain many possibilities for use of 3D formats. Accompanied with powerful PDM systems which manage whole workflows, 3D viewer facilitates archiving and quick retrieval of product data in each phase of product creation processes. In general, the benefit of 3D lightweight formats is extra high if products consist of large assemblies or have a high variance or engineering changes are frequent [10]. Finally, standardized applications cause cost savings in administration and training (see Sect. 21.4).

In the context of Concurrent Engineering, 3D lightweight formats facilitate synchronous operations by rapid sharing of data and quick access to data provided from different sources. Hereby, they provide a unique platform for information sharing between broad user circles. As a PDM system is recognized as a digital backbone (see Chap. 16), the viewer fuelled by 3D lightweight formats can be understood as windows to this backbone (see Sect. 23.3.2).

11.7.1 *Automotive*

The automotive industry was, followed closly by the aerospace industry, the main driver for advanced engineering visualization (see Chap. 21) due to product complexity and variety, short product cycles and high investment volumes, where each new technology promises a quick return of investment. Beginning with the first draft, in almost every phase of product creation processes, the need for visualization exists, in particular for 3D product representation. Powerful DMU tools use JT as standard input (see Sect. 13.5).

11.7.1.1 Design in Context

In term of CAD, modern passenger cars consist of huge assemblies with many thousands of parts. Such assemblies are difficult for visualization because the loading time alone can already last many hours. Nevertheless, the design work with huge assemblies is inevitable in case of design in context (DiC) where designer needs the entire environment to put his creation into place preventing interferences and collisions. The use of JT for this purpose is indicated in the case that the CAD system is able to handle the JT data in a similar way as native data.

The DiC process is one of the most complex processes which could be based on JT- and structure information (in future e.g. STEP AP 242XML) coming from various 3D CAD systems. These formats can be combined with native 3D CAD data which is often needed and beneficial in design processes (no media break). In today's DiC processes, often the exchange of geometrical change information is needed. The DiC process often refers to interface regions of (e.g. externally created) parts. Information located in these regions (e.g. geometrical elements) can be used for DiC-processes. The main categories in which this information can be divided are the geometrical information and the technological information.

Geometrical information is often used to define geometrical elements in surrounding parts on customer/OEM side. Therefore information such as diameters, curves or supplemental geometry is used. In case technological information is needed a broad range of information is thinkable in interface regions of parts e.g. mechanical and electrical information. These interface related feature information can be transferred out of the sourcing 3D CAD system into JT-data using e.g. PMI or in future using a combination of JT and XML-based accompanying dataformats such as STEP AP242XML for the transfer of additional information [10, 14].

In the Fig. 11.12, the yellow shaded parts are the externally created parts of a diesel engine which are, therefore, candidates to be supplied as JT files [15]. This means that in this case, native inventory 3D CAD data stored in the PDM system—successfully used in today's development processes—have the same content as potential JT files. Details of this analysis are described in the following chapter.

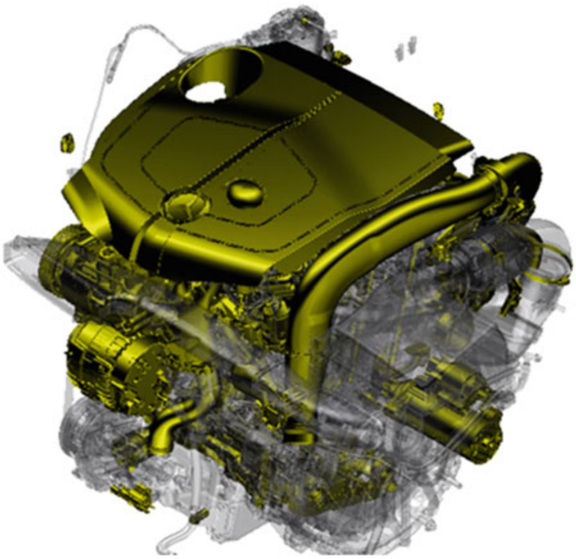


Fig. 11.12 Potential for use of JT for DiC at Daimler

11.7.1.2 Data Analysis Gather Status Quo

Analysis of inventory data is one of the key prerequisites for successful JT usage in collaboration processes [10]. Data that has been created on OEM side and data which come from a supplier, e.g. by an automatic evaluation of supplier part drawings or PDM information, have to be distinguished.

The focus of the data analysis is to find out which information is contained in inventory data. Examples may include the identification of parametric, design history, relations and formula.

After such several general assumptions have to be made in order to use JT (cost) beneficial while knowing the limitations of the JT data format (e.g. missing parametric and design history). Therefore the most relevant criteria which have to be fulfilled are whether or not the

- design lead for the part is on supplier side;
- the part is used in series production and not in an early development stage;
- kinematic information does not need to be exchanged in combination with JT;
- part does not contain advanced electrical harness information.

Keeping this in mind still a huge variety of parts fits in this newly defined part range. The reason for this is that in today CAD-based processes the information included in the parts and the differentiation of the design lead is handled in similar way. This is also validated by first analysis results of the Daimler AG [15].

The most important benefits are listed in the following (Fig. 11.13):

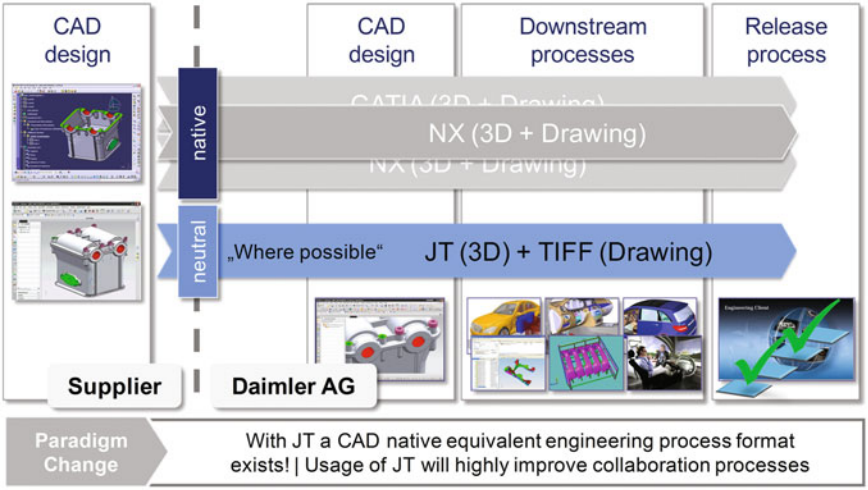


Fig. 11.13 Paradigm change in usage of CAD formats at Daimler [15]

- CAD system independency: Suppliers are independent of used CAD system or (in Daimler's case) NX version.
- Reduction of data preparation efforts: JT data helps to reduce data preparation efforts on supplier side in company with native CAD-data.
- Downstream process needs: JT data is able to fulfil Daimler R&D and downstream processes needs for a defined part range.
- Wider range of suppliers: JT data enables a wider range of possible suppliers worldwide.
- ISO Standard: JT data is ISO Standard—ISO 14306 and will be accompanied with STEP AP 242 XML.

Having a closer look at the above listed advantages of JT usage it seems obvious that the supplier is becoming independent of the used CAD system because for all relevant CAD systems (at least in automotive and aerospace industry) there are translation tools available that convert native CAD data of all currently used CAD systems in a very good quality to the JT data format [16]. The quality of the translators is benchmarked once a year at the PSI JT Translator Benchmark (see Sect. 11.5).

Furthermore, one of the biggest advantages of the use of JT is the fact that the JT data format does not contain the huge complexity that todays native CAD models do, having a complex OEM-given model structure as a basis. The JT data format has just to be created in a way the exchanging partner accepts it [17]. Having this in mind, a common setting of the JT models was found in the automotive area driven by the PSI in the JT Workflow Forum.

11.7.1.3 Downstream Processes in House

As a further step towards the productive usage of JT in automotive downstream processes, pilot projects are established. Volkswagen decided to use JT for internal downstream processes to eliminate the need for drawings [13, 18]. This application is limited to the use case "single parts" and shall be subsequently extended to the full assemblies to avoid the usage of huge JT monolithic files. Involved are the downstream processes enumerated with 1 and 3 in Fig. 11.5 in particular the wide area of industrial engineering, [19] where huge potential benefits were identified.

In case of Daimler, where a transition and data migration is running from CAD systems CATIA V5 to NX, JT is fully accepted as a CAD native equivalent engineering process format for many processes (Fig. 11.13). Use of engineering drawing is not excluded yet, but it shall be also replaced by JT as much as possible. Exclusive use of native CAD formats remains in core design areas where specific process needs or applications exist (e.g. parametric relationships, knowledge-based engineering (KBE), kinematics, flexible parts, advanced wire harnessing).

11.7.1.4 Supplier Integration

Supplier integration is becoming more and more important due to the fact that value addition is more and more connected to the supplier's side [20] (see Sect. 21.2). On the other hand, dealing efficiently with complex heterogeneous IT landscapes is one of the key factors in the successful supplier integration itself [21] (see Chap. 7). Having this in mind, it is obvious to use standardized data exchange formats such as STEP (ISO10303) and JT (ISO14306) to overcome these challenges (see Sect. 21.7).

In order to fulfill OEM needs (e.g. in case of data quality) it has to be ensured that the JT and the accompanying structure formats are created properly. In a first step the configuration settings of the JT translator have to be adjusted to the OEM needs. Therefore, the aforementioned PSI JT content harmonization workgroup (JTCH) has created a first best practice document [9]. In the second step, the OEM specific data preparation has to be done. By removing the parametrics during the translation from native CAD to JT, an important step in intellectual property protection is executed (see Sect. 18.6). Therefore, the Daimler JT supplier package (JTSP) is available for all Daimler suppliers to facilitate the handling of JT data [6].

The Daimler JTSP offers the opportunity to edit the JT internal attributes and to prepare XML based structure formats to meet the OEM requirements. The third step in the process landscape is the data exchange to the OEM. In case of the Daimler AG the Odette file transfer protocol (OFTP) based SWAN data exchange system is used.

The final step of this process is the data import into the Daimler PDM system Smaragd. At this point in time, the externally created JT datasets are released and available to all Daimler downstream processes such as Viewing, DMU or DiC [15].

In the following the supplier integration use case is presented (Fig. 11.14). The use case can be distinguished in

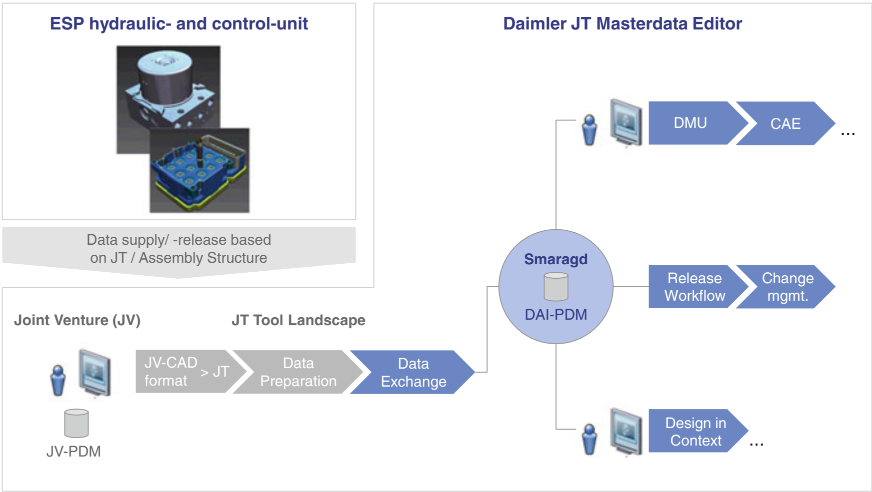


Fig. 11.14 Usage of JT in partner integration [18]

- Early phase (conceptual phase with limited requirements)
- Late phase (with increased requirements).

In a first step the early phase of the use case will be shown and first requirements will be derived.

A supplier sends a monolithic JT file (enveloping surface and connection points, no inner structure) for the entire work package with all needed metadata to the OEM. JT data has to be quality checked against a pre-defined check profile. The quality check will ensure the completeness of masterdata, geometry (e.g. open faces), colours etc. Every time JT contents were created—even if they were created on the fly—they have to be quality checked (maybe in batch process) to prevent unnecessary issues in the following processes. Units of measurement (weight, length), surface area, material thickness, volume, material information, center of gravity and moment of inertia must be provided in the package. Because of long term archiving reasons the same JT file version should be used as in ISO 14306 (currently version 2012).

In the later phase, a monolithic file would not be precise enough and presumably difficult to handle. Thus, a supplier is requested to send a “per Part” JT (enveloping surface and connection points, no inner structure, but with further fixing and context relevant data) with all needed metadata to the OEM. In this case, additional data on product structure is needed e.g. by using a corresponding STEP file (as demonstrated in Sect. 21.7).

One of the most challenging projects was the Daimler/Continental JT pilot project by exchanging an ESP Electronic Control Unit. In the following the process landscape (use case: supplier to OEM) of the Daimler/Continental JT pilot project is shown in Fig. 11.14.

Meanwhile, the full CAD data exchange by using JT is going to be adopted as a base CAD translation technology by supplier portals like OpenDESC.com (www.opendesc.com) [22]. Like each other CAD translation, the use of JT sets high requirements to the data creation processes, methods and corresponding data quality [23].

11.7.1.5 3D-Based Shape Search

For several reasons (e.g. reduction of production costs), designers of new products preferably use existing parts instead of creating new ones (see Chaps. 14 and 17). Systematic reuse of existing parts and modules sets high requirements to entire product data management (see Chap. 16) and requires appropriate measures and tools [24]. To find parts which fit the new needs, a geometrical search within former designs could be very helpful. For such purpose there are search engines like Geolus from Siemens PLM which is a geometry-based search engine for both single and multi-CAD environments. The search base is a collection of similarity vectors which are calculated for the existing parts during their archiving in JT after the approval procedure.

The search runs by loading an existing CAD model within the design environment or using a catalog of standard parts or a new 3D sketch. In the next step the geometric search is conducted by starting the Geolus software. Using JT records a search is performed within seconds by geometrically similar components (Fig. 11.15). The results are listed and give the user a detailed analysis and selection of each matching component. The user can refine his search until the desired variant is found and, therefore, use an existing component for a new or variant design. This saves development time and costs, and demonstrates the benefits of 3D data-based searches.

11.7.1.6 Multi-CAD

The usage of various CAD systems by different partners or even by individual in-house departments is a daily business within challenging product development processes. A CAD system is selected and introduced once, and shall be used during a long period. Change of CAD system is an expensive, tedious process which shall be considered carefully and avoided, if possible [25]. Nevertheless, certain flexibility in use of a multi-CAD environment must be preserved. In this context, JT is used as a kind of backbone format to serve different partners or within supplier integration processes. Instead of complex, native conversion between different CAD formats, JT is used for DiC within a multi-CAD environment. Distinction must be made between non-hybrid and hybrid DiC. The key feature of hybrid usage is that the context data based on JT can be loaded into the native CAD environment, displaying native CAD and JT data in parallel. JT data is not converted to a CAD-internal format.

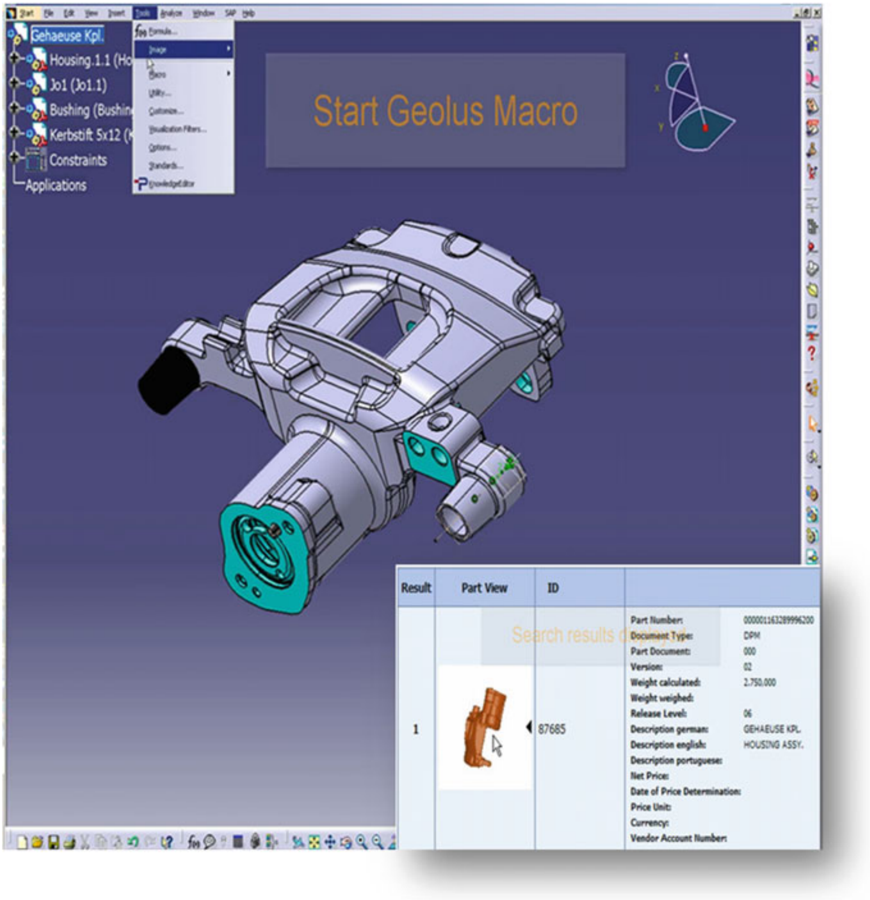


Fig. 11.15 3D-based shape search at Continental [12]

Depending on a given set of parameters, e.g. distance to viewer, JT contents are visualized in respective quality via different LODs or exact geometry.

In the case of multi-CAD based on non-hybrid DiC, the JT context data can be imported into the native CAD environment. JT contents are then converted to an internal CAD representation.

Within a regular multi-CAD process a design engineer creates a particular product structure and geometry, which is to be used as context by another design engineer. The exported JT files, including PMI, such as dimensions and tolerances includes selective features as well, in order to be extracted by the receiving design engineer.

The receiving design engineer loads JT geometry into the native CAD system, which is then displayed hybrid to native CAD geometry, with the possibility to switch between different resolutions of tessellated geometry, or to exact geometry.

This allows handling of large assemblies. Native CAD geometry may continuously be modified, with the addition of now being able to position CAD data in relation to certain JT content, such as edges, vertices, faces, or auxiliary geometry, e.g. point, axis or plane. Exact geometrical references, e.g. curves, are also possible. New PMI can be added, associating native CAD or JT-based geometry. Going beyond typical PMI like surface finish, selective advanced features, e.g. a drill hole with parametric definition could be selected and displayed with the extensive content. Optionally, the parametric definitions can be altered. When creating a technical drawing, new PMI and loaded JT content are included as well.

An important prerequisite for a functioning multi-CAD environment is the quality assurance. JT-data has to be quality checked against a defined check profile. The quality check will ensure the completeness of master data, geometry (e.g. open faces), colors etc. Every time JT contents were created—even if they were created on the fly—the have to be quality checked. This may be done in a batch process [23].

Through this JT based multi-CAD approach, a paradigm shift is taking place. With JT, a CAD native equivalent engineering process format arises. The usage of JT highly simplifies collaboration processes. JT enables part design in multiple CAD systems and DMU in mixed assemblies. As benefits interiors of supplier parts are hidden, data volumes get smaller and CAD data exchange becomes independent from CAD systems.

Within future implementations, it is also imaginable to consider communication from changes within JT back to the CAD model. For instance new PMI provided based on JT are then transferred back to the original CAD model, which is respectively updated. Likewise, if geometry is changed in the JT, this is communicated back to the primary CAD model. As a result native CAD geometry would be positioned correctly in relation to JT context geometry. New PMI and feature definitions would be updated in the original CAD model.

11.7.1.7 Archiving

For archiving of engineering data, it is generally necessary to take into account the exact representation of the data, including all metadata and PMI information. The latter are of importance in particular for the drawingless manufacturing, in which drawings are substituted by 3D digital models, and 2D drawings are not available as archiving medium any more.

For the long-term archiving of engineering data it is the main requirement that all relevant information is stored in a format that can be read independently of a specific IT infrastructure and even after a long period of time. After JT was standardized as ISO 14306, whereby its unique definition as well as its long-term maintenance is ensured, it can be used for archiving too. JT data can be extracted automated or manual from the authoring CAD and PDM systems and be used for archiving purpose. The archiving process steps are illustrated within Fig. 11.16.

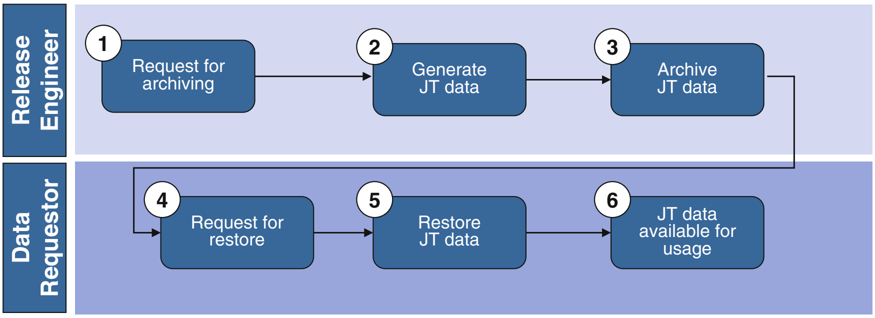


Fig. 11.16 Archiving process steps [10]

Once again, data quality is of enormous importance for the archiving use case. Besides the verification of the data quality, the validation of the JT files against the native 3D CAD files has to be ensured [23].

The most important are the

- trouble-free combination of data from various authoring systems,
- ensuring that the data can be accessed after a long period of time and
- to cover all product related information.

Advantage of using JT for the 3D part of the archive is the comprehensive JT file format specification, the long list of applications available and the established quality assurance measures.

11.7.2 Machinery

Design and production of machines is a broad field determined by low series, light supplier involvement, high reuse of existing, carry-over or standard parts and relatively simple geometric properties of parts. Many machines have long dimensions or are generally very large. For such reasons this industry has been looking for alternatives to paper drawing for a long time.

11.7.2.1 Drawingless Manufacturing

Nowadays, the development of new products takes place almost continuously in 3D. However, derived 2D drawings are still considered the authoritative documents for manufacturing, quality assurance, assembly and other downstream processes. These 2D documents often contain valuable technical manufacturing information, such as product and manufacturing information (PMI), material properties bills of material, and annotations. However, 2D drawings discover fundamental deficits in transparency and concurrency and are going to be replaced by appropriate 3D data.

The aim of drawingless processes is to base decision processes, etc. on 3D documents without derived 2D drawings. For storing any kind of geometry information like simplified models or fully detailed designs the JT format is used. To handle additional meta data different accompanying formats are meaningful: STEP AP242 XML for any kind of assembly structure or PDM information. For further meta data like release status, change history of technological information the PDF format is highly appropriate. Beneficial with this approach is that any combinations are possible through the use of 3D PDF documents. For instance the geometrical product and manufacturing information can be covered by JT or the PDF internal 3D representation.

A 3D PDF document containing 3D geometry models as well as additional meta data is shown in Fig. 11.16.

3D PDF containers bring together all the information in the product development process into a single file. A compact container may contain all the 3D models, PMI, and other related documentation necessary to integrate external manufacturing partners into the product design development but also manufacturing process (Fig. 11.17).

The main benefits for the use case drawingless manufacturing are

- less manual effort creating and revising drawings and other paper-based documentation,
- a single container for all information from the product development process and
- the elimination of distributing and storing paper drawings.

11.7.2.2 Purchase Process with Request for Quotation

The purchase unit controls the central tendering processes in an enterprise by using request for quotation. In case of technical goods seamless integration of technical (e.g. 3D models) and administrative content can be highly useful. By using 3D PDF technology all necessary information for an inquiry can be collected in one document (RFQ) and, therefore, frequently exchanged between requester and bidder. Used in conjunction with the PLM integration modules, 3D PDF technology allows technical procurement staff to automate the integration of 3D models of parts to be manufactured externally in their requests for quotation, enabling bidders to calculate their offers more easily [11]. With this in mind, a form-based solution can accelerate the RFQ process as part of a customer project. This allows supplier data to be read automatically from the ERP system together with the purchase requisitions and inserted into a PDF form template that is then sent to all the eligible bidders. In the reverse direction, data from incoming offers can be extracted automatically and transferred back to the ERP system for the purposes of bid comparison.

The use of these ‘intelligent’ 3D PDF documents, which are filled automatically with the current 3D models and 2D information from the back-end systems, also speeds up the creation of sales and service documentation and their updates when

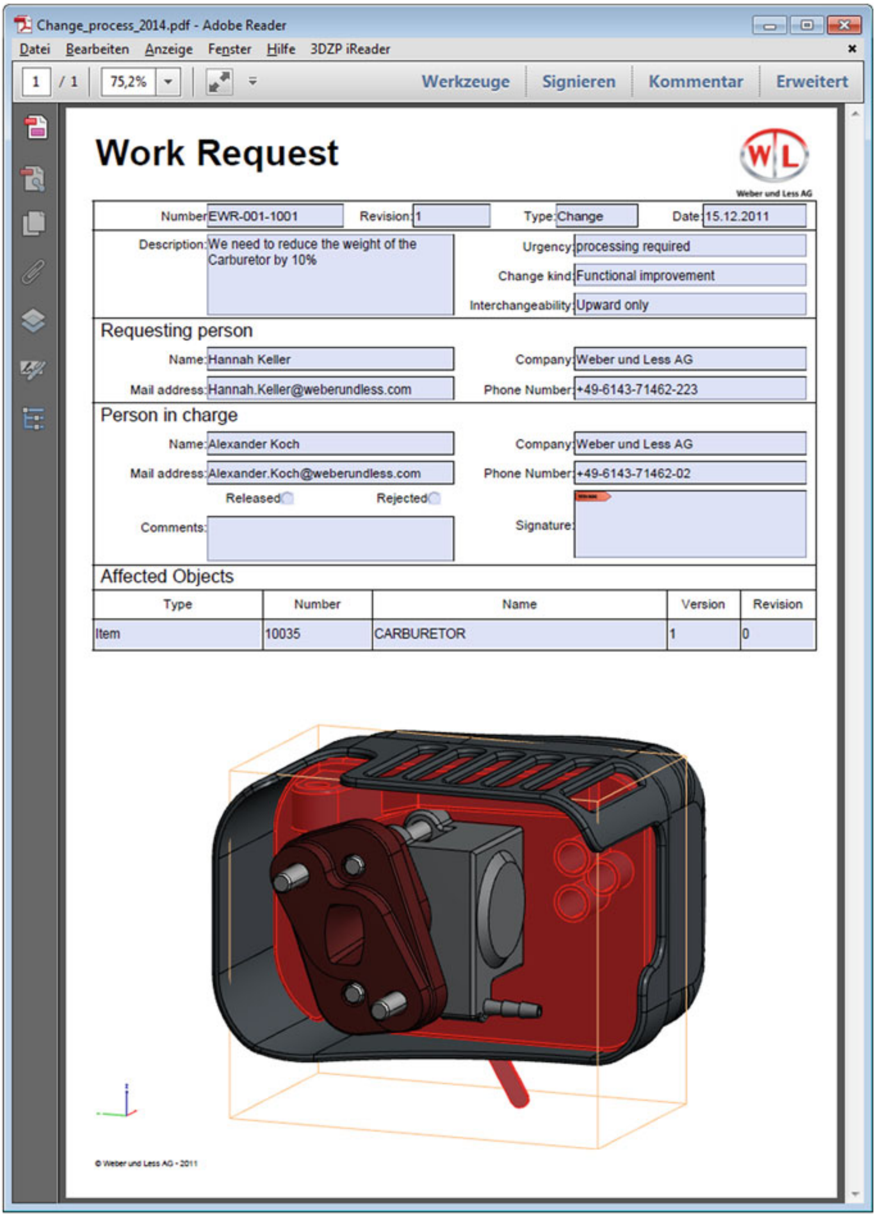


Fig. 11.17 3D PDF document [7]

products are modified. The rapid provision of such documentation presents a huge challenge to companies with global operations in particular. One of the ways in which they can use the 3D PDF technology is to augment spare parts lists with 3D information, thus facilitating the search for the requisite spare parts by service engineers working on the customer's premises.

Such a solution can be fully integrated into the Adobe Digital Enterprise Platform (ADEP) which provides right management (Sect. 18.6.3.3). The end-user is still working in his well-known Acrobat Reader environment [8].

11.8 Future Directives

As broad commercial exploitation of IT standards pre-requisites their adoption by main software vendors, requirements have to be harmonized and prioritized carefully with precisely defined use cases. Many use cases address requirements regarding JT, STEP AP242 XML, or both formats. It is necessary to systematize the detailed interaction of both formats. For instance, the usage of product structure is possible with both formats with multiple options. As a consequence, best practices from the user perspective are a prerequisite. Based on the elaborated use cases, the derived requirements have be detailed and balloted by the users. One important mechanism is external referencing to ensure the seamless interaction of JT with the accompanying format STEP AP242 XML. For instance, external reference mechanisms are a prerequisite for kinematics. The first prototypes for the transmission of assemblies including kinematics between different CAD systems have already been implemented successfully. Thus, the first neutral, standard-based CAD data exchange between development partners is possible. Nevertheless, an implementation in the current CAx applications and converters is still pending in many cases.

Within the current JT standard ISO 14306:2012 90 % out of the 200 singular format requirements elaborated by the users were already fulfilled. As a consequence, most of the 30 defined use cases can be implemented expecting their full functionality [24]. The remaining format requirements are addressed to the next JT version V2. The standardization roadmap of JT and STEP AP242 is illustrated in Fig. 11.18.

Besides the standardization of the visualization formats and the functionality of the applications, there is a strong need to harmonize the usage of JT. Reasoned by the complexity and multifold of the data format, there are many degrees of freedom regarding the content of JT files. For instance, it is mandatory to find agreement for naming of user defined attributes, the accuracy of the applied LOD or the usage of specific product structure options. Furthermore, a clear distinction between JT and STEP AP242 XML must be made. The user has to define which information should be stored within which format. By way of example, the storage of meta data is possible within JT but also within STEP AP242 XML. These harmonization activities have to be done in-house but even more across the enterprise.

Further exploitation of 3D PDF technology is coordinated by the 3D PDF Consortium, an international community founded by 15 companies interested in collaboration of dynamic 3D data through PDF files [27]. PDF will continue to evolve and be managed through the ISO process—specifically ISO32000.

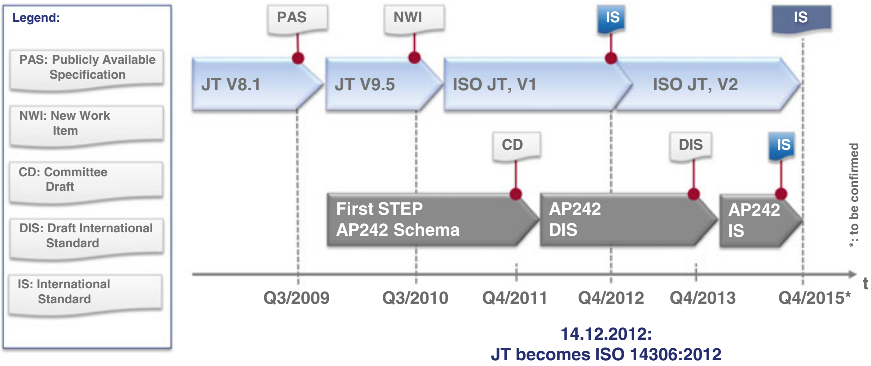


Fig. 11.18 The schedule of JT and STEP AP 242

Recognizing the value of harmonized implementation (see Sect. 11.3.2), the 3D PDF Consortium has initiated the 3D PDF Implementor Forum in February, 2014. It collaborates closely with the Association for Intelligent Information Management (AIIM) which is responsible for the on-going development of the specification for the PRC data format and PDF/E-2. The 3D PDF Consortium promotes 3D PDF adoption through demonstrating best practices and generating awareness of the power of 3D-enabled PDF to solve a multitude of communication and collaboration challenges across various industries.

11.9 Conclusions and Outlook

This chapter has addressed the state of the art activities for establishing new 3D lightweight formats (JT, 3D PDF) as a universal process format in engineering domains. To harness the full benefits of 3D data, the manufacturing industries require simple solutions for the efficient use of IT technology, including consistency in processes. There are further requirements for lightweight 3D formats for the visualization and downstream processes, complementary formats in order to exchange meta data, structure data and kinematics data as well as open and standardized formats to reduce total cost of ownership and to minimize dependency of single vendors [26, 28, 29]. At this time, commercial formats JT and 3D PDF fulfil most of criteria and can be included in many industrial workflows [30, 31].

After these very promising, aforementioned achievements, the activities on the subject of lightweight visualization based on JT and 3D PDF have already reached a good level of maturity, encouraging the wide range of companies and users. The capabilities of JT are proven by benchmark which runs every year. Furthermore, its benefits were evaluated in many applications. The sum of software adopters increases continuously.

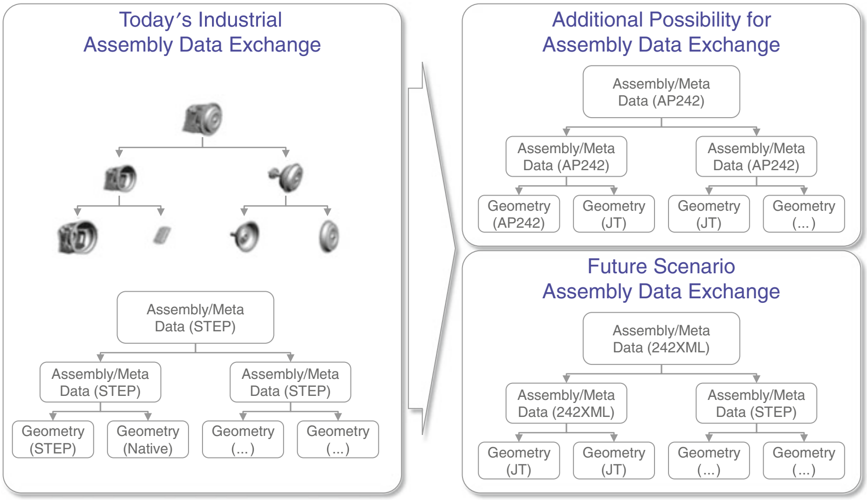


Fig. 11.19 Assembly data exchange with JT and STEP AP 242

Further integration with the accompanying format STEP AP242 is the major forthcoming task for JT development (Fig. 11.19). Although first applications creating or using STEP AP242 XML as an out- or input format are already available today, the wide usage will be announced in the next years [32]. The future data exchange process will adopt JT as well as STEP AP242, ensuring the exchange of whole complex products at all stages of product and process development. Further development is preserved by international bodies which include implementer fora.

References

1. Carlson W (2003) A critical history of computer graphics and animation. The Ohio State University. http://design.osu.edu/carlson/history/lessons.html. Accessed 15 Feb 2014
2. Luebke D, Reddy M, Cohen JD, Varshney A, Watson B, Huebner R (2003) Level of detail for 3D graphics. Morgan Kaufmann, San Francisco
3. Shirley P, Marschner S (2009) Fundamentals of computer graphics, 3rd edn. CRC Press, Boca Raton
4. Anderl R, Malzacher J (2007) Fast data and information flows with collaborative product visualization. ProductData J 1:46–48
5. NN (2005) Digital engineering visualization, SASIG, SASIG guideline. Source http://www.aiag.org/source/Orders/index.cfm?search=D-21
6. NN (2007) Collaborative product visualisation—general issues and use case description, Frankfurt, VDA, VDA Empfehlung 4966. Source http://www.prostep.org/fileadmin/freie_downloads/Empfehlungen-Standards/VDA/VDA_4966_Collaborative-Product-Visualization_1.0.pdf
7. Pfalzgraf P, Pfouga A, Trautmann T (2012) Cross enterprise change and release processes based on 3D PDF. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches

for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering, 2013. Springer, London, pp 753–763
8. Fröhlich A (2013) 3D formats in the field of engineering—a comparison, PROSTEP AG, Darmstadt. http://www.pdfgenerator3d.com/nc/en/product/white-paper.html. Accessed 15 March 2014
9. Biahmou A, Fröhlich A, Stjepandic J (2009) Universelle Formate für Visualisierung und Konvertierung in der Engineering-Kollaboration. International Conference GO 3D, Rostock, Aug 30–31
10. Handschuh S (2011) Wertextrahierende Nutzung von offenen leichtgewichtigen Datenformaten in automobilen Kollaborations- und Entwicklungsprozessketten. PhD thesis, Technische Universität Kaiserslautern
11. NN (2010) Applying JT—guidance for using JT in Practice Version 2, Whitepaper ProSTEP iViP e.V. http://www.prostep.org/de/mediathek/veroeffentlichungen/white-paper-studien.html. Accessed 15 March 2014
12. Beckers R, Fröhlich A, Stjepandic J (2010) Anwendung und Potenziale universeller Visualisierungsformate, 9. Paderborner Workshop "Augmented & Virtual Reality in der Produktentstehung", Paderborn. http://www.transmechatronic.de/uploads/tx_vitramemberadmin/literature/Anwendung_und_Potenziale_universeller_Visualisierungsformate.pdf. Accessed 15 March 2014
13. Handschuh S, Dotzauer R (2012) JT on the path of enlightenment—report of the JT working groups. ProSTEP Symposium, Hamburg, 10 May 2012
14. Gerhardt F (2010) Supporting virtual product engineering processes by integrating a neutra, lightweight and CAD-derived data format. PhD thesis, Technische Universität Kaiserslautern
15. Handschuh S (2013) Driving JT at Daimler: JT-Strategy, potentials, roll-out in cooperation with PLM2015, Daimler EDM CAE Forum, Stuttgart, 10–11 July 2013
16. Katzenbach A (2012) Potentials of JT in the automotive industry, In: JT open international conference 2012, Barcelona, 13 Sep 2012
17. Stjepandic J, Emmer C, Fröhlich A (2013) Advanced engineering visualization with standardized 3D formats. In: Bernard A, Rivest L, Dutta, D (eds.) Product lifecycle management for society, Proceedings of the 10th IFIP WG 5.1 international conference, 2013. Springer, Heidelberg, pp 584–595
18. Handschuh S, Dotzauer R, Fröhlich (2012) Standardized formats for visualization—application and development of JT. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering, 2013. Springer, London, pp 741–752
19. Röhl H, Fröhlich A (2012) Einsatz des JT-Formates in der Digitalen Fabrik. VDI Kongress "Digitale Fabrik", Regensburg, 5–6 Nov 2012
20. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Sys Manag 6(4):307–323
21. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Sys Manag 7(2):116–131
22. Bondar S, Potjewijd L, Stjepandić J (2012) Globalized OEM and Tier-1 processes at SKF. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering, 2013. Springer, London, pp 789–800
23. Bondar S, Ruppert C, Stjepandić J (2014) Ensuring data quality beyond change management in virtual enterprise, Int J Agile Sys Manag 7(3–4):304–323
24. Maranzana R (2013) 3D data mining for part and information re-use in a PLM context. Global Product Data Interoperability Summit, Chandler, USA,Sept 9–12
25. Katzenbach A (2012) Methodik zum Austausch eines CAD Systems in einem Großunternehmen. International Conference „Entwerfen - Entwickeln – Erleben 2012", Dresden, July 3

26. Vetterman S (2013) Why are you still talking? Putting JT and STEP into practice. ProductData J 2:17–20
27. 3D PDF Consortium. http://www.3dpdfconsortium.org/pdf-standards-info.html. Accessed 15 March 2014
28. Dineva E, Bachmann A, Moerland E, Nagel B, Gollnick V (2014) New methodology to explore the role of visualisation in aircraft design tasks: an empirical study. Int J Agile Sys Manag 7(3–4):220–241
29. Casera S, Kropf P (2010) Collaboration in scientific visualization. Adv Eng Inform 24 (2010):188–195
30. Shen Y, Ong SK, Nee AYC (2008) Product information visualization and augmentation in collaborative design. Comput Aided Des 40(2008):963–974
31. Qiu ZM, Kok KF, Wong YS, Fuh JYH (2007) Role-based 3D visualisation for asynchronous PLM collaboration. Comput Ind 58(2007):747–755
32. Ungerer M (2013) Managed model based 3D engineering. ProductData J 2:9–11

Chapter 12
Reverse Engineering

Goran Šagi, Zoran Lulić and Ivan Mahalec

Abstract One of the most time-consuming aspects of creating 3D virtual models is the generation of geometric models of objects, in particular if the virtual model is derived (digitized) from a physical version of the object. A variety of commercially available technologies can be used to digitize objects at the molecular scale but also multi-storey buildings or even planets and stars. The process of 3D digitizing basically consists of a sensing phase followed by a rebuild phase. The sensing phase collects or captures raw data and generates initial geometry data, usually as a 2D boundary object, or a 3D point cloud. Sensing technologies are based on tracking, imaging, and range finding or their combination. The rebuild phase is internal processing of data into conventional 3D CAD and animation geometry data, such as NURBS and polygon sets. Finally, in most cases, the digitized objects must be refined by using the CAD software to gain CAD models of optimal quality which are needed in the downstream processes. Leading CAD software packages include special modules for such tasks. Many commercial vendors offer sensors, software and/or complete integrated systems. Reverse engineering focuses not only on the reconstruction of the shape and fit, but also on the reconstruction of physical properties of materials and manufacturing processes. Reverse engineering methods are

G. Šagi (✉) · Z. Lulić · I. Mahalec
Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb,
Ivana Lučića 5, 10000 Zagreb, Croatia
e-mail: goran.sagi@fsb.hr; info@adolo7.hr
URL: http://adolo7.hr/

Z. Lulić
e-mail: zoran.lulic@fsb.hr
URL: http://adolo7.hr/

I. Mahalec
e-mail: ivan.mahalec@fsb.hr
URL: http://adolo7.hr/

G. Šagi · Z. Lulić · I. Mahalec
Adolo 7 d.o.o., Ivana Lučića 5, 10000 Zagreb, Croatia

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_12

applied in many different areas, ranging from mechanical engineering, architecture, cultural heritage preservation, terrain capture, astronomy, entertainment industry to medicine and dentistry.

Keywords Reverse engineering · Scanning methods · Shape reconstruction · Feature reconstruction · Innovative design · Intellectual property protection

12.1 Introduction

Engineering is the process of designing, manufacturing, assembling, and maintaining products and systems. There are two types of engineering: forward engineering and reverse engineering. Forward engineering (FE), or engineering design, is a process of creating a *new* part or a complete product, applying imagination, creativity and originality. Reverse engineering (RE) is a process of *duplicating* an *existing* part, assembly, or product, generally without any technical documentation. Chikofsky and Cross [1] defined RE as “the process of analysing a subject system to—identify the system’s components and their interrelationships and—create representations of the system in another form or at a higher level of abstraction”. Wang [2] described RE as “a process of measuring, analysing, and testing to reconstruct the mirror image of an object or retrieve a past event. It is a technology of reinvention, a road map leading to reconstruction and reproduction. It is also the art of applied science for preservation of the design intent of the original part”. Pham and Hieu [3] defined RE “as a process of analysing an object or existing system (hardware and software) to identify its components and their interrelationships and to investigate how it works to redesign or produce a copy without access to the design from which it was originally produced”. The focus of this chapter is on the acquisition of the shape of an industrial product.

Many modern machines were invented with inspiration from nature, or reinvented through reverse engineering based on what was observed in nature. The airplane is one of the most noticeable examples. One of the widely cited reverse engineering examples in the military is the Soviet Tupolev Tu-4 bomber. During World War II, three battle-damaged U.S. B-29 Superfortress bombers made emergency landings in the Soviet Union. Although most airplanes can be distinguished from one another by their respective characteristics, the similarity between the general characteristics of the B-29 and the Tu-4 bomber (Fig. 12.1) has led to a conclusion that the Tupolev Tu-4 was a replica of the B-29 [2].

Nowadays, a three-dimensional geometric model is widely used in engineering. It is an intermediate representation shared among the participants in the product creation process and it contains the semantic information needed in the singular steps (e.g. simulation parameters, materials, tolerances and manufacturing entities). If the subject is an existing product whose digital model is unavailable, it is necessary to first rebuild it with appropriate reverse engineering techniques. For many

Fig. 12.1 Boeing B-29 superfortress bomber (*left*) and tupolev Tu-4 bomber (*right*)

years, reverse engineering methods have found wide application in manufacturing, industrial design and reverse innovative design (RID).

Answers to the questions why RE is needed and what is its role in Concurrent Engineering can be found in Sect. 12.2, accompanied with related studies. The RE process, data acquisition process and the most common non-contact scanning methods are described in Sect. 12.3. Importance of material characteristics, durability and life limitations are given in Sect. 12.4. Tools for RE, especially software tools for data processing, are presented in Sect. 12.5. Section 12.6 presents applications of the RE methods in different areas: automotive industry, railways, machinery, architecture, archaeology and medicine. Ethical and legal issues considering RE are discussed in Sect. 12.7, followed by brief conclusions and an outlook.

12.2 Background and Related Work

RE allows the duplication of an existing part by capturing its physical dimensions, features and material properties. The challenge for RE is to reproduce this part with an equivalent or preferably better functionality at lower costs. RE usually offers a good value for money only if the items to be reverse engineered reflect high development costs or will be reproduced in large quantities. But even if it is not cost effective, RE is sometimes the best choice for engineers who have to carry out a task of producing, for example, a part which is indispensable and crucial to the system. Furthermore, RE allows the study into an unknown or malfunctioning system to be carried out in order to enhance its efficiency and reliability. As to why the use of RE might be the best solution for accomplishing an engineering task, Raja [4, 5] and other authors quote the following:

- The original part or its design data are no longer available, but a customer needs the product for repair.
- Creating data to refurbish or manufacture a part for which there are no CAD data, or for which the data have become obsolete or lost.
- Inspection and Quality Control—Comparing a fabricated part to its CAD description or to a standard item.
- Some bad features of a product need to be improved, e.g. excessive wear.

- Analysing the good and bad features of competitors' products.
- Exploring new avenues to improve product performance and features.
- Creating 3D data from an individual model or sculpture for animation in games and movies, or to create, scale, or reproduce artwork.
- Fitting clothing or footwear to individuals and determining the anthropometry of a population.
- Architectural and construction documentation and measurement.
- Generating data to create dental or surgical prosthetics, tissue engineered body parts, or for surgical planning.
- Documentation and reproduction of crime scenes.

12.2.1 Reverse Engineering in Concurrent Engineering

Although reverse engineering techniques have been applied in very different areas, they are applied in a similar manner. RE not only helps to rapidly reproduce a competitive product, but can also be used to study the behaviour of an existing system. In the forward development process, Müller et al. [6] proposed to apply simultaneously reverse engineering and forward engineering to achieve a more complete understanding of the system. In this way, RE reduces the number of faults in the development process and speeds up significantly the development of our own new products. Performances, accuracy and speed of various RE techniques depend crucially on the applied scanning method (Table 12.1).

12.2.2 Related Work

The area of reverse engineering is developing in several directions: development of new methods and technologies, improvement of tools for data acquisition (hardware) and data processing (software) in terms of the accuracy and acceleration of the process itself, and usage of RE methods in new applications. Several important papers deal with the development in each of these areas.

A comprehensive overview of the state of the art of 3D sensing techniques and devices, their applications in a wide range of measurement problems in industry, cultural heritage, medicine and forensics can be found in [8]. The authors present an overview of techniques and sensors for the optical 3D measurement of surfaces and evaluate different approaches to highlight which method can be used and what are its main applications. They also give an insight into the results achieved in the mentioned applications. The overview of systems proposed in this paper yields a number of conclusive remarks. The first one concerns the cost of the equipment for 3D acquisition. The second remark concerns the fact that the use of 3D acquisition is not a trivial task: the systems are still rather complex to use and need skilled

Table 12.1 Three RE types based on the end-use applications and technical requirements

RE types	Objectives and technical requirements
Industrial reverse engineering	**Aim**: Reconstruction of 3D geometrical models of physical objects for engineering design, CAD-CAM-CAE-CNC-RP-RP&T Product development Quality control and dimensional inspection **Typical object size**: from 200 × 200 × 200 mm to 500 × 500 × 500 mm **Accuracy requirement**: Typically from ±20 to ±50 μ In mould and tooling and in micro-manufacturing: up to (1 to 5) microns In ship building and aeronautic industry: quite flexible, depending on the size of objects and their functions
Artistic and architectural reverse engineering	**Aim**: 3D geometrical modelling and control of objects **Field**: Topography, architectural and facade measurements, as-built surveying, archaeology and cultural heritage documentation and city modelling Fashion and arts: 3D art modelling, portrait sculpturing and prototyping 3D graphics and animations: virtual reality, games and films **Object size**: from 10 × 10 × 10 mm to large topographic areas **Accuracy**: Low in comparison with industrial RE. Outside appearance, the general shape and forms of objects have priority over accuracy
Medical reverse engineering (MRE)	**Aim**: Medical application development and research. It is normally involved in using patient data or biomedical objects to reconstruct 3D models of anatomical structures and objects of interest for the development of different medical products, applications, and biomedical research **Accuracy** (depending on specific applications): For the personalised cranio-maxillofacial implants, bio-models and training models, the accuracy requirement is not very stringent compared to industrial RE, i.e. it is up to hundred(s) of microns For surgical tools and functional implants such as spine, hip and knee implants, the accuracy requirement is very stringent

Derived from [3, 7]; *CAE* computer-aided engineering, *CNC* computer numerically-controlled

personnel to operate them. The third remark, which is crucial from the metrologist's viewpoint, is the need for norms to guarantee the traceability of 3D measurements to recognised standards. As 3D acquisition is rather new, these norms are not yet mature. The final remark concerns the fact that 3D systems, in many complex metrological issues, may not represent the solution to the problem when used alone. The concurrent use of combinations of contact and non-contact systems, including 3D systems, may be required for a full metrological solution. The most common RE methods are described in Sect. 12.3, and applications of the methods in different areas in Sect. 12.6.

An analysis of a digitization system on the basis of its accuracy, effectiveness and the quality of distribution of points and triangular meshes in the field of reverse engineering can be found in [9]. The actual experimentation with simple or complex objects and different materials yields results that, in some cases, refute the effectiveness of those systems. In order to help in choosing a digitization system on the basis of its accuracy and the quality of distribution of points and triangular meshes, the authors compared a few digitization techniques. It is shown that measurements on real calibrated pieces and with different materials give greater uncertainties than those given by the manufacturers. According to the authors, the quality of the scanning system has been divided into: the accuracy of digitization, the digitization of the piece, the distribution of points, the roughness of the mesh, the mesh of edges and holes without meshing.

In [10], the authors evaluate recent advances in data acquisition and processing, and provide an overview from a manufacturing perspective. Success of generating a virtual representation of a physical object from a dataset of point clouds relies on reliable algorithms and tools. Whereas 3D scanners have become powerful, the performances of the corresponding software tools are perceived as unsatisfactory. Commercial 3D modelling tools lack the ability to deal with large amounts of data. End users often wish to automatically process a wide range of objects, possibly from a variety of data capture devices with different characteristics, to produce models in a variety of representations and accuracies. Many effective methodologies have been developed to solve various problems involved in data acquisition and processing. But without a sound background knowledge of mathematics and computer science, it is often hard to understand theoretical fundamentals of the methodologies. The authors give an overview of software tools for data processing through data filtering, data registration and integration, feature detection, 3D reconstruction, surface simplification and segmentation. Also, the authors list available 3D scanners, their manufacturers and stand-alone 3D data processing software tools. Tools, especially software tools for data processing, are presented in detail in Sect. 12.5.

Surface reconstruction from point clouds is fundamental in many applications. A brief overview of surface reconstruction methods and a literature review are available in [10]. In the topic of surface reconstruction and speeding up the process of data processing, there is still a lot of room for improvement. Many researchers are involved in this specific area. In [11], the authors proposed a surface meshing method capable of dealing with a great variety of surfaces such as those which are closed or not, orientable or not, uniformly sampled or not, with non-manifold intersections or without. The current implementation of the proposed method is roughly as fast as other recent popular methods. A new high-performance method for triangular mesh generation based on a mesh-growing approach is proposed in [12]. The performance of the proposed method has been compared with the performance of reference method with applications of the mesh-growing approaches to some benchmark point clouds and artificially noised test cases. The results show that the proposed method is competitive in terms of tessellation rate, quality of the generated triangles and produced defectiveness. In [13], the authors proposed a

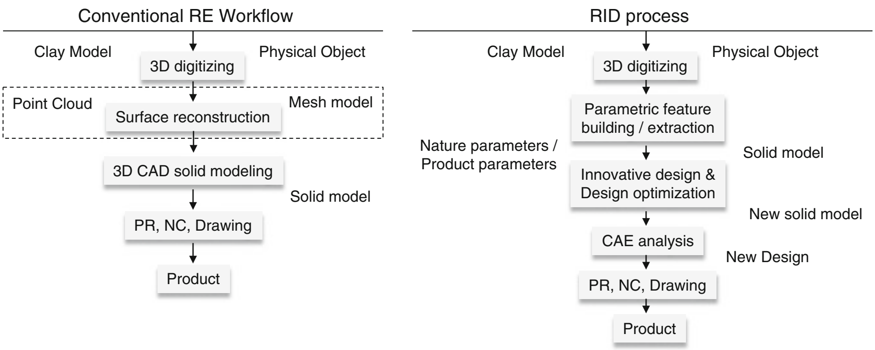


Fig. 12.2 Workflow of conventional RE versus RID [14]

robust algorithm to surface reconstruction. The robustness of the proposed method and some other methods is demonstrated on several examples with noise and invalid data and with dense point clouds. In spite of limited application with respect to the size of the input point cloud, both in running times and memory use, the proposed method is proved to be a versatile tool for surface reconstruction.

Application of reverse engineering methods to new areas is shown on the example of new design methodology called RID proposed in [14]. RID methodology comprises advanced design methodologies that facilitate the acquisition of design knowledge and creative ideas for later reuse. RID is an integrated digital design methodology incorporating digitizing, modelling with shape and product definition parameters, CAE analysis-based product optimization and rapid prototyping (RP). Figure 12.2 shows a comparison between the workflows of conventional RE and RID. The core of RID is the feature based parametric solid model constructed from scanned data, for analytically shaped models as well as models with freeform shapes. For analytically shaped models, features with natural definition parameters will be extracted with remaining freeform shapes fitted. Since freeform shapes stand at the centre of RE historically, the ability to generate parametric models with product definition parameters from models with freeform shapes is essential.

In [15], the authors present a new technique for reconstructing a single shape and its deformation (non-rigid motion) from a temporal sequence of point clouds from real-time 3D scanner data. In addition, the authors give a brief overview of related work in reconstructing correspondences of time variant geometry, review the related work in the area of deformation modelling and compare it to the proposed technique. The authors apply the technique to several benchmark data sets, increasing significantly the complexity of the data that can be handled in comparison to the previous work, while at the same time improving the reconstruction quality.

Review and classification of methods for the acquisition of surface geometry or volumetric descriptions of objects or phenomena with complex optical characteristics (transparent, specular, etc.) can be found in [16]. While the 3D acquisition of opaque surfaces is a well-studied problem, transparent, refractive, specular and

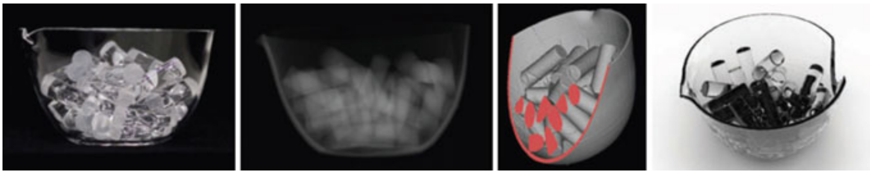


Fig. 12.3 Fluorescent immersion range scanning. A photograph of an acrylic glass object (*left*), a direct volume rendering of a recovered voxel model (*middle*, *left*), a cut-away iso-surface view (*middle*, *right*) and a realistic rendering (*right*) [16]

potentially dynamic scenes pose difficult problems for acquisition systems. The acquisition of digital models of such objects is far from being a solved problem. This report is providing a reference for and an introduction to the field of the transparent and the specular object reconstruction. Figure 12.3 shows an example of transparent object scanning.

12.3 Methods of Reverse Engineering

When the shape of an object should be reconstructed without access to its design, a RE process has to be applied. The selection of RE technology depends on various factors: size of the object, its complexity, its material (hard or soft), its finish (shiny or dull), its geometry (internal or external), the required accuracy, etc.

12.3.1 Reverse Engineering Process

The RE process consists of three phases: object scanning, point processing and generation of 3D geometrical model (Fig. 12.4). The first phase, i.e. data acquisition, is a crucial part of RE. In this phase RE hardware is used to collect geometric data that represent a physical object. The outputs are point clouds or 2-D cross-sectional images that define the scanned object geometry. RE data produced by RE hardware are transformed into a 3D geometric model by using the RE software. The final outputs of this data processing are a polygon mesh or a NURBS mesh (*n*on-*u*niform *r*ational *B*-*s*plines mesh). Polygon models, usually in the STL, VRML, or DXF format, are used for RP, laser milling, 3D graphics, simulations and animations. NURBS surfaces or solids are used in computer-aided design, manufacturing and engineering applications (CAD-CAM-CAE).

1. 3D Scanning: In recent years, an enormous number of various scanning technologies for capturing the shape of a physical object have been made available at reasonable prices. The selection of appropriate tools and techniques has become very challenging. The starting point for all of them is the acquisition of

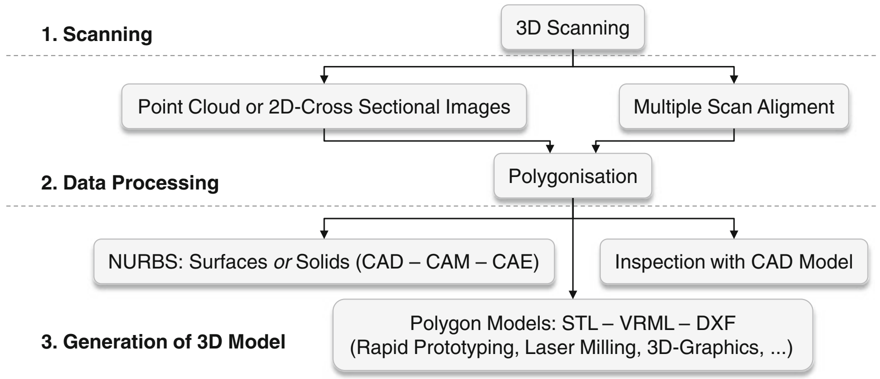


Fig. 12.4 The generic process of reverse engineering. Adapted from [4]

a set of X-Y-Z coordinates in space, called point clouds, in one of the convenient output formats. The clouds are then processed in the second phase of RE process to provide an applicable output for the re-creation of the scanned object.

2. Data Processing: This phase involves importing the point cloud data, reducing the noise in the data collected and reducing the number of points [4]. These tasks are performed using a range of predefined filters. It is extremely important that the users have very good understanding of the filter algorithms so that they know which filter is the most appropriate for each task. This phase also allows us to merge multiple scan data sets. Sometimes, it is necessary to take multiple scans of the part to ensure that all required features have been scanned. This involves rotating the part; hence each scan datum becomes very crucial. Multiple scan planning has a direct impact on the point processing phase. Good datum planning for multiple scanning will reduce the effort required in the point processing phase and also avoid introduction of errors from merging multiple scan data. A wide range of commercial software is available for point processing. The output of the point processing phase is a point cloud data set in the most convenient format.
3. Generation of 3D Model: The generation of 3D surface or solid CAD models from point data is probably the most complex activity within RE because potent surface fitting algorithms are required to generate surfaces that accurately represent the three-dimensional information described within the point cloud data sets. Most CAD systems are not designed to display and process large amounts of point data; as a result, new RE modules or discrete software packages are generally needed for point processing [4]. The RE software allows the user also to compare the two different data sets. This process is very useful for inspections of manufactured parts. In such cases, the designed CAD model is imported by appropriate software and overlaid with the scanned point cloud data set of the machined part.

12.3.2 Data Acquisition

The taxonomy of hardware used for collecting geometric data of an explored object is shown in Fig. 12.5. There are two main non-destructive technologies for RE data acquisition: contact and non-contact.

When contact technology is used for scanning, tactile measurement machines (Fig. 12.6) have to touch the object of interest to measure its geometry [4]. In such a case either the object or the measurement probe could be damaged. Some commercial coordinate measuring machine (CMM) systems claim to be non-contact devices, but they still require a measurement probe to be quite close to the point of measurement, just not touching it. On the other hand, non-contact optical systems make measurements at some standoff distance. Furthermore, if the temperature of the surface is too hot or too cold, the heat transfer could damage the measurement probe. With a touch probe, a CMM (or a user) must carefully select a measurement path that properly covers the surfaces of an object but that avoids wedging the probe into tight spaces. The CMM must use a path that covers the object and yet obeys the physical constraints imposed by the interaction between the object and the probe. In practice, however, due to their disadvantages, optical 3D scanning systems are less used than CMMs. One of the most significant features is accuracy. Ultrahigh accuracy CMMs work in the 1–2 μm range, and more moderate CMMs (in terms of cost) in the 10–20 μm range. Computer vision methods cannot compete —as of yet—with these levels where most systems operate in the submillimeter range of accuracy. The trend, however, indicates that CMMs have plateaued. Only a few microns difference in accuracy can result in more than a €100 K increase in cost. On the other hand, computer vision research indicates that greater accuracy is yet to come with more precise lasers and higher resolution imaging sensors.

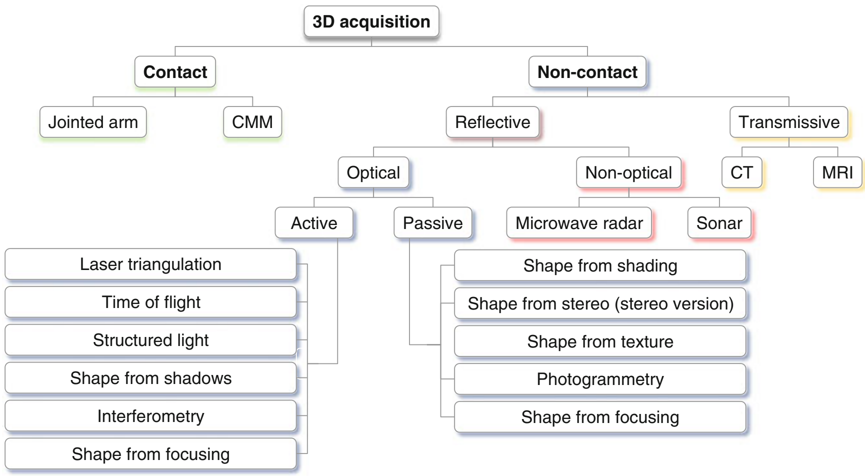


Fig. 12.5 A taxonomy of scanning hardware according to [3, 8, 17]

Fig. 12.6 Multi-axis tactile measurement machines: Spin Arm M, 7-axis articulated measurement arm and Crysta Apex, 3-axis CNC CMM (both from Mitutoyo)

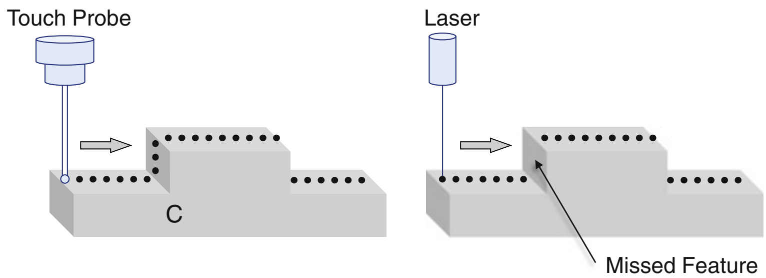


Fig. 12.7 Contact versus non-contact sensing: the scanning probe cannot touch the sharp inside corner *C*, and the laser beam missed vertical walls parallel to laser axis [4]

Contact scanners employ contact probes that automatically follow the contours of a physical surface (Fig. 12.7). In the current marketplace, contact probe scanning devices are based on CMM technologies, with a tolerance range of +0.01 to 0.02 mm. However, depending on the size of the part scanned, contact methods can be slow because each point is generated sequentially at the tip of the probe. Tactile device probes must deflect to register a point; hence, a degree of contact pressure is maintained during the scanning process. This contact pressure limits the use of contact devices because soft, tactile materials such as rubber cannot be easily or accurately scanned.

Contact methods use sensing devices with mechanical arms, CMMs and computer numerical control (CNC) machines, to digitize a surface. There are two types of data collection techniques employed in contact methods:

1. Point-to-point sensing with touch-trigger probes and
2. Analogue sensing with scanning probes.

In the point-to-point sensing technique, a touch-trigger probe is used that is installed on a CMM or on an articulated mechanical arm to gather the coordinate points of a surface [18]. A manually operated, articulated mechanical arm with a touch-trigger probe allows multiple degrees of freedom (DOF) of movement to collect the measurement points (Fig. 12.6, left). A CMM with a touch-trigger probe can be programmed to follow planned paths along a surface. A CMM provides more accurate measurement data compared to the articulated arm. However, the limitation of using a CMM is the lack of number of DOF so that a CMM cannot be used to digitize complex surfaces in the same way as an articulated arm. In analogue sensing, a scanning probe installed on a CMM or CNC machine is used (Fig. 12.6, right). The scanning probe provides a continuous deflection output that can be combined with the machine position to derive the location of the surface. When scanning, the probe stylus tip contacts the feature and then moves continuously along the surface, gathering data as it moves. Therefore, throughout the measurement, it is necessary to keep the deflection of the probe stylus within the measurement range of the probe. The scanning speed in analogue sensing is up to three times faster than in point-to-point sensing. The more advanced CMM systems allow operators to upload a CAD model of the object and then the CMM uses this model for the path planning strategy. The CMM will analyse the CAD model to identify critical points and regions.

Non-contact Scanners: A variety of non-contact scanning technologies available on the market capture data with no physical part contact [4]. Non-contact devices use lasers, optics and charge-coupled device (CCD) sensors to capture point data. Although these devices capture large amounts of data in a relatively short period of time, there are a number of issues related to this scanning technology:

- The typical tolerance of non-contact scanning is within ±0.025 to 0.2 mm.
- Some non-contact systems have problems generating data describing surfaces, which are parallel to the axis of the laser beam (Fig. 12.7).
- Non-contact devices employ light within the data capture process. This creates problems when the light impinges on shiny surfaces, and hence some surfaces must be prepared with a temporary coating of fine powder before scanning.

These issues restrict the use of remote sensing devices to areas in engineering where the accuracy of the information generated is secondary to the speed of data capture. However, as research and laser development in optical technology continue, the accuracy of the commercially available non-contact scanning device is beginning to improve (Table 12.2).

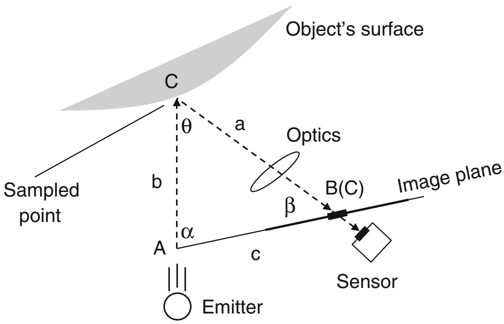
Table 12.2 Advantages and disadvantages of the contact and non-contact techniques [7]

Technique	Advantages	Disadvantages
Contact	High accuracy Low-costs Ability to measure deep slots and pockets Insensitivity to colour or transparency	Slow data collection Distortion of soft objects by the probe
Non-contact	No physical contact Fast digitizing of substantial volumes Good accuracy and resolution for common applications Ability to detect colours Ability to scan highly detailed objects where mechanical touch probes may be too large to accomplish the task	Possible limitations for coloured or transparent or reflective surfaces Lower accuracy

12.3.3 Optical Scanning Methods

There is a remarkable variety of 3D optical techniques, and their classification, as given in Fig. 12.5, is not unique. In this section, some of the more prominent approaches are briefly explained and compared (Table 12.3).

Active optical devices are based on an *emitter*, which produces some sort of structured illumination on the object to be scanned, and a *sensor*, which is typically a CCD camera and acquires images of the distorted pattern reflected by the object surface [17]. In most cases the depth information is reconstructed by triangulation (Fig. 12.8), given the known relative positions of the emitter-sensor pair. The emitter can produce coherent light (e.g. a laser-beam) or incoherent light; in both cases, a given light pattern (point-wise, stripe-wise or a more complex pattern) is projected on the object surface. Different technologies have been adopted to



$$\theta = 180° - (\alpha + \beta)$$

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\theta}$$

Fig. 12.8 A scheme of a typical optical scanner, where the 3D positions of the sampled points are computed by triangulation given the sampled point projection B(C) on the sensor plane and the known relative position/orientation of the emitter and the sensor [17] produce the structured light pattern: laser emitters, custom white light projectors (which can filter light by means of a glass slide with a stripe pattern engraved via photo-lithography), low cost photographic slide projectors and finally digital video projectors.

Passive methods reconstruct a 3D model of an object by analysing the images to determine coordinate data [3]. It is similar to (active) structured-light methods in its use of imaging frames for 3D reconstruction; however, in passive methods, there is no projection of light sources onto the object for data acquisition. The typical passive methods are shape from shading and shape from stereo.

Laser triangulation method is a technique which uses the law of sine to find the coordinates and distance of an unknown point by forming a triangle with it and two known reference points [2]. In Fig. 12.8, A and B are the two reference locations given by the camera and the sensor locations, and C is the location of the object point of interest. The distance from A to B can be measured as c, and the angles α and β can also be measured. Following the law of sine the distances a and b can be calculated. The coordinates of A and B are known, then the coordinate of C can also be calculated. The same principle is employed in various other scanning methods.

Time-Of-Flight (**TOF**) method uses the radar time-of-flight principle [8]. The emitter unit generates a laser pulse, which impinges onto the target surface (Fig. 12.9). A receiver detects the reflected pulse, and suitable electronics measures the roundtrip travel time of the returning signal and its intensity. Reflective markers must be put on the target surfaces. The measurement resolutions vary with the range. For large measuring ranges (15–100 m), time-of flight sensors give excellent results. On the other side, for smaller objects, about 1 m in size, attaining 1 part per 1,000 accuracy with time-of-flight radar requires very high speed timing circuitry, because the time differences are extremely short. In many applications, the technique is range-limited by allowable power levels of laser radiation, determined by laser safety considerations. Additionally, time-of-flight sensors face difficulties with shiny surfaces, which reflect little back-scattered light energy except when oriented perpendicularly to the line of sight.

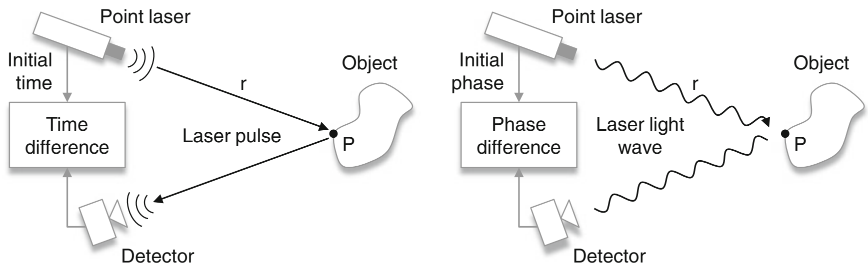


Fig. 12.9 Time-Of-Flight system (TOF) (*left*) measures the time required for a laser pulse to travel to and return from an object. Continuous wave system (*right*) is a variation on the TOF method: distance is computed by comparing the phase shift between an emitted wavelength and the received light [18]

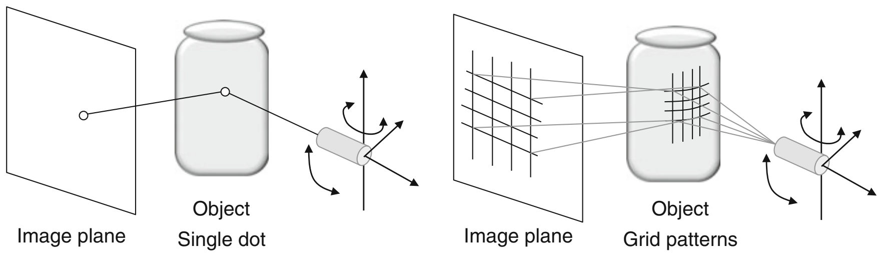


Fig. 12.10 Different light patterns used in structured-light techniques [3]

Structured-light systems project a predetermined pattern of light onto the object, at a known angle. An image of the resulting pattern, reflected by the surface, is captured, analysed and the coordinates of the data point on the surface are calculated by using a triangulation method. The light pattern can be (i) a single point; (ii) a sheet of light (line); and (iii) a strip, grid, or more complex coded light (Fig. 12.10). The CCD camera, the object and the light source form the triangulation geometry (Fig. 12.11). The accuracy of these methods is primarily a function of the camera resolution and secondarily of geometric dimensions and illumination precision. System geometry and illumination are not as critical. Thus, structured-light systems offer a more practical solution than passive stereographic systems in achieving the accuracy necessary for an RE system.

The most commonly used pattern is a sheet of light, generated by fanning out a light beam. To improve the capturing process, the light pattern containing multiple strips is projected onto the surface of an object. The strips must be coded to enable the recording without ambiguity. Structured-light systems have the following strong advantages compared to laser systems: (i) the data acquisition is very fast (up to millions of points per second), (ii) colour texture information is available, (iii) structured-light systems do not use a laser and because of that they have the advantage of being inherently eye-safe. These features have resulted in favouring structured-light systems for digitizing images of human beings.

Moiré interferometry is used to measure tiny deformations of solid bodies, caused by mechanical forces, temperature changes, or other environmental changes [20]. It has been applied for studies of composite materials, polycrystalline materials, layered materials, piezoelectric materials, fracture mechanics, biomechanics, structural elements and structural joints. It is practiced extensively in the microelectronics industry to measure thermally induced deformation of electronic packages. Moiré interferometry combines the simplicity of geometrical moiré with the high sensitivity of optical interferometry, measuring in-plane displacements (Fig. 12.12). It is characterised by a list of excellent qualities. Moiré interferometry has a proven record of applications in engineering and science.

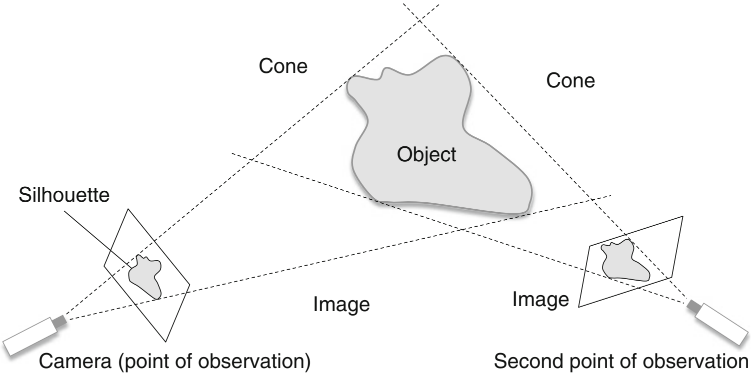


Fig. 12.11 A simple approach to recover the shape of an object: shape from contours (or silhouettes). The silhouette and the point of observation for each view form a cone containing the object. The intersection of multiple cones is an estimate of object shape. Shape from contour techniques, however, fail at recovering object concavities [19]

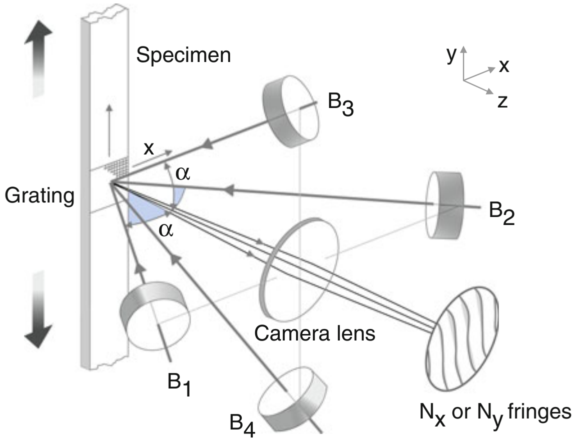


Fig. 12.12 Schematic diagram of four-beam moiré interferometry [20]

12.3.4 Other Non-contact Scanning Methods

Computer Tomography (**CT**) allows three-dimensional visualization of the internals of an object. It provides a large series of 2D X-ray cross-sectional images taken around a single rotational axis [3]. By projecting a thin X-ray or Y-ray beam through one plane of an object from many different angles and measuring the amount of radiation that passes through the object along various lines of sight, a cross-sectional image for the scanned surface is reconstructed (Fig. 12.13). CT is widely used for medical applications; however, it has been extended and adapted to

Table 12.3 Comparison of optical range imaging techniques [8]

Technology	Strenth	Weakness
Laser triangulators	Relative simplicity Performance generally independent of ambient light High data acquisition rate	Safety constraint associated with the use of laser source Limited range and measurement volume Missing data in correspondence with occlusions and shadows Cost
Photogrammetry	High data acquisition rate Simple and inexpensive High accuracy on well-defined targets	Computation demanding Sparse data covering Limited to well defined scenes Low data acquisition rate
Time-of-flight	Medium to large measurement range Good data acquisition rate Performance generally independent of ambient light	Cost Accuracy is inferior to triangulation at close ranges
Structured light	High data acquisition rate Intermediate measurement volume Performance generally dependent of ambient light	Safety constraints, if laser based Computationally middle-complex Missing data in correspondence with occlusions and shadows Cost
Stereo vision	Simple and inexpensive High accuracy on well-defined targets	Computation demanding Sparse data covering Limited to well defined scenes Low data acquisition rate
Interferometry	Sub-micron accuracy in micro-ranges	Measurement capability limited to quasiflat surfaces Cost Limited applicability in industrial environment
Moiré fringe range contours	Simple and low cost Short ranges	Limited to the measurement of smooth surfaces
Shape from focusing	Simple and inexpensive Available sensors for surface Inspection and microprofilometry	Limited fields of view Non-uniform spatial resolution Performance affected by ambient light
Shape from shadows	Low cost Limited demand for computing power	Low accuracy
Texture gradients	Simple and low cost	Low accuracy
Shape from shading	Simple and low cost	Low accuracy

a wide variety of industrial and 3D modelling tasks. Today, high-resolution X-ray CT and micro CT scanners can resolve details as small as a few tens of microns, even when imaging objects are made of high-density materials. It is applicable to a wide range of materials, including rock, bone, ceramic, metal and soft tissue.

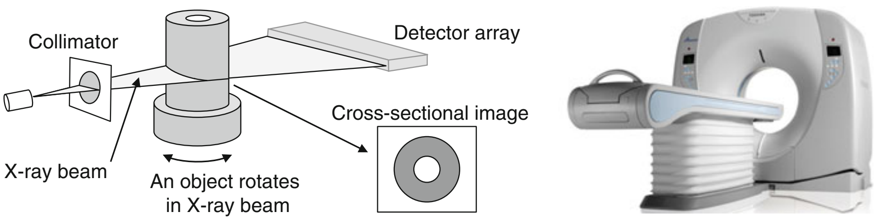


Fig. 12.13 Working principle of a CT scanner [3] and CT scanner for medical applications

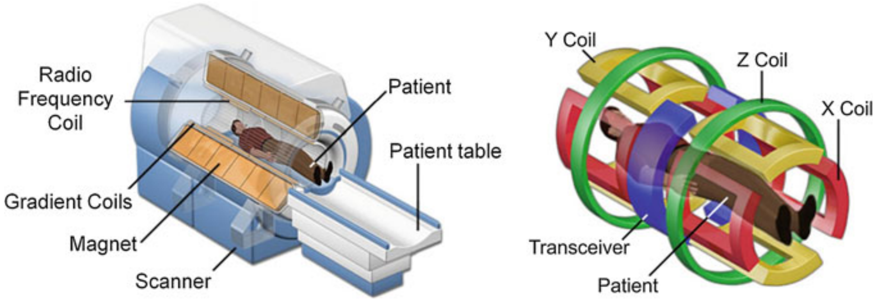


Fig. 12.14 Magnetic resonance imaging (MRI): scanner and gradient magnets [21]

Fig. 12.15 The ATOS measurement (*above*, *left*), the 3D models made up of the STL files (*above*, *right*), a 3D model of the racing vehicle for the Croatian Dakar Rally Team (*down*) [23]

Magnetic Resonance Imaging (**MRI**) is a state-of-the-art imaging technology that uses magnetic fields and radio waves (Fig. 12.14) to create high-quality, cross-sectional images of the body without using radiation [3]. When hydrogen protons in the human body are placed in a strong magnetic field, by sending in (and stopping) electromagnetic radio-frequency pulses, the protons emit signals. These signals are collected and processed to construct cross-sectional images. Compared to CT, MRI

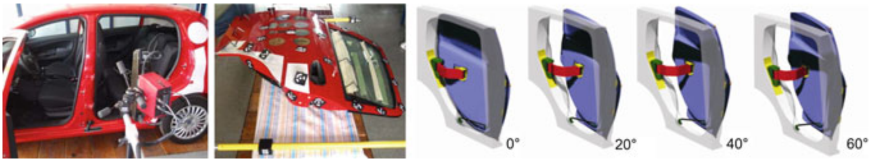


Fig. 12.16 The TRITOP/ATOS measurement (*left*), a CAD model of the new rear door opening mechanism (*right*) [23]

Table 12.4 Comparison of performances of various non-contact scanning technologies. Data extracted from [3]

Technology	Accuracy	Speed (points/s)
Laser triangulation	8 μm–5 mm	6,666–256,000
Structured light	0.025–0.070 mm	6,666–442,368
Time-of-flight	0.025–25 mm	1,750–200,000
Interferometry	0.5 μm–0.6 mm	850–1,857,142
Computer tomography	5 μm–0.25 mm	18,641–154,202

gives superior quality images of soft tissues such as organs, muscle, cartilage, ligaments and tendons in many parts of the body.

CT and MRI are powerful techniques for medical imaging and reverse engineering applications; however, they are the most expensive techniques in terms of both hardware and software for data processing (Table 12.4).

12.4 Material Characteristics, Durability and Life Limitation

Material characteristics are the cornerstone for material identification and performance evaluation of a part made by using reverse engineering [2]. The determination of relevant material characteristics and their equivalency requires a comprehensive understanding of the material and the functionality of the part that was made of this material. To convincingly argue which of the mechanical, metallurgical and physical properties are the most relevant material properties that need to be evaluated in a reverse engineering project, the engineer has to carry out at least the following elaboration:

1. Property criticality: Explain how critical this relevant property is to the part's design functionality.
2. Risk assessment: Explain how this relevant property will affect the part performance, and what the potential consequence will be if this material property fails to meet the design value.
3. Performance assurance: Explain what tests are required to show the equivalency to the original material.

Mechanical properties include ultimate tensile strength, yield strength, ductility, fatigue endurance, creep resistance and stress rupture strength. *Metallurgical properties* refer to the physical and chemical characteristics of metallic elements and alloys, such as the alloy microstructure and chemical composition. *Physical properties* usually refer to the inherent characteristics of a material (i.e. density, melting temperature, heat transfer coefficient, specific heat, electrical conductivity, etc.).

Many mechanical components have life limits in their service due to the deterioration of their durability over time. Although it is more technically challenging to reverse engineer a life-limited part, market demands and higher profit margins provide strong incentives for their reproduction using reverse engineering. The life cycle of a part is determined either by the total load cycles the part has experienced or by the total time period the part has been placed in service. The reverse engineered parts are expected to maintain the same level of safety attributable to the integrity of materials and machine functionality. A mechanical component usually fails due to excessive elastic deformation, excessive plastic deformation, fracture, environmental effects or a combination of these factors. The prevention of part failure requires full knowledge of material characteristics, loading condition and service environment. A thorough understanding of the part design functionality and operation is critical for reproducing an equivalent mechanical component using reverse engineering.

Fatigue is a dynamic and time-dependent phenomenon. When a component is subject to alternating stresses repeatedly, it fails at a much lower stress than the material yield strength due to fatigue. Most mechanical failures are related to dynamic loading; therefore, the safety assessment in fatigue life plays a critical role in reverse engineering.

The performance of a reverse engineered part compared to its original equipment manufacturer (OEM) counterpart is vitally critical to the success of a reverse engineering project. The performance of these parts is usually evaluated based on three primary criteria: engineering functionality, marketability and safety. From an engineering functionality perspective, part performance is judged based on its structural integrity and system compatibility [2].

12.5 Tools for Reverse Engineering

Although the domain of RE is very broad, as mentioned in Sect. 12.3.1, a conventional RE process involves the following three steps [14]:

1. 3D scanning of physical projects, typically generating a point cloud.
2. Data processing such as noisy data removal, registration, sampling, smoothing, topology repair and hole-filling.
3. Surface reconstruction from mesh or point cloud by direct surface fitting or surface reconstruction through curves such as section curves and feature lines.

Data acquisition and data processing include both hardware and software tools and the results of RE are usually surfaces that need to be imported into a 3D CAD software. A hardware system acquires point clouds or volumetric data by using established mechanisms or phenomena for interacting with the surface or volume of an object of interest. There are many types of 3D scanning or data acquisition systems available, as shown in Sect. 12.3.2, which differ in their characteristics such as accuracy, speed, working volume, environmental operating constraints, reliability, cost, etc.

A software system processes raw point clouds or volumetric data and transfers them into a virtual representation of the object such as surfaces and features. One of the critical tasks of vision-based manufacturing applications is to generate a virtual representation and its success relies on reliable algorithms and tools. Processing of raw scanned data or data cleaning is very important since curves and reconstructed surfaces are based on the mesh model. Data processing and surface reconstruction is the centre piece of a RE process. The interpretation of raw data to a required computer model is a complicated process, and it involves the following typical issues [10]:

Data Filtering: Raw data include noise, distorted and invalid data caused by the hardware system and/or the environment. The acquired data must be filtered to eliminate invalid data. Point data can be invalid due to many reasons, often caused by reflectivity of surface elements, objects in the background, moving objects, atmospheric effects, bright objects, etc. The elimination process often has to be done interactively since no automatic method can foresee all possible causes. The acquired data also may be filtered to reduce the level of noise caused by the precision of the data acquisition system or to reduce the number of points in a dense area.

Data Registration and Integration: Registration and integration are needed for two different purposes, first, the combination of several point clouds taken from different observation points and second, the referencing of the object in a global coordinate system. A vision device can capture the surface facing the device in the field of view. Therefore, multiple views are needed to acquire data over the entire surface, and the data from different views have to be integrated. The registration is used to determine the transformation of data from two different views so that data can be integrated under the same coordinate system. Integration is the process of creating a single surface representation from the sample points of two or more range images.

Surface (3D) Reconstruction: Surface reconstruction from point clouds is fundamental in many applications. Using the raw point clouds or volumetric data acquired from an unknown surface, an approximation of the surface can be constructed and used to compare it with CAD models or for surface-based automated programming. Reconstruction methods can be classified into two types: the computational geometry approach focuses on the piecewise-linear interpolation of unorganised points and defines the surface as a carefully chosen sub-set of the Delaunay triangulation in a Cartesian coordinate system, and the computer graphics approach focuses on the visual quality of the resulting model without constraining the surface to interpolate the sampled points.

Surface (Data) Simplification and Smoothing: A compact approximation of a shape can reduce memory requirements and accelerate data processing. It can also accelerate computations involving shape information, such as finite element (FE) analysis, collision detection, visibility testing, shape recognition and display. Simplification is useful to make storage, transmission, computation and display more efficient.

Data Segmentation: Segmentation involves the partitioning of a given image into a number of homogeneous segments in such a way that the union of any two neighboring segments yields a heterogeneous segment. Segmentation refers to a process for extracting the selected regions of interest from the rest of data using automated or manual techniques. Data filtering and segmentation are two different aspects of the same problem. A good filtering process should distinguish between a set of significant regions and the border between them. Such a filtering process assumes, implicitly, that the segmentation is known.

Feature Detection: Feature detection is used to recover a high-level geometric description from the lower-level geometric representation of a part. Defects or some basic elements on a surface can be dealt with as features. Examples of such features include size, position, contour measurement via edge detection and linking, as well as texture measurements on regions. Feature detection is used to identify defects with certain features or validate if the acquired data fit a specific feature.

Data Comparison: A reference model is usually available for data comparison. Data comparison calculates the derivations or differences between the physical model and the reference model. It can be applied to inspection, surface control, or CAD model comparison. For example, (i) in feature detection, point clouds can be used to measure geometric elements such as planes, cylinders, circles, spheres and boundaries; and (ii) in monitoring and control, as-designed and as-built models are compared so that the deviation (average error), tolerance and distribution can be evaluated.

Many methodologies have been developed to solve various issues involved in data processing and some of them are mentioned in [10]. Available software tools for data processing include an extensive collection of modules for different purposes ranging from scanner control to 3D modelling.

In recent years, the RE process has not only been used for scanning and converting data into a 3D surface, but also into solid parts. RP is often used in industry due to its capability of creating 3D parts with complex geometries. To fabricate a part by using RP processes directly from a representation in the form of point cloud data, the necessity of integrating data processing in RE and RP is comprehensively established. One of the methods for the direct generation of RP models from arbitrarily scattered cloud data can be found in [22].

12.6 Use Cases

Application of the RE methods is widespread in various fields of human activity. Some of the applications are illustrated by the cases to be discussed next. The RE methods are often integrated with other methods, thus representing a multidisciplinary approach to problem solving.

12.6.1 Automotive

In the automotive industry, the RE methods are for instance used in the following areas: redesign, reconstruction and dimensional control. The following two cases are similar examples of vehicle reconstruction with different final uses. In order to make the required design changes, a 3D model of original vehicle parts had to be created and measured. The surface of such complex objects and specific points on the object can be digitised using various optical measuring systems. In the following tasks, presented in [23], two measuring systems were used: TRITOP (an optical 3D CMM) and ATOS (an optical 3D scanner).

One of the tasks in the development of a racing vehicle for the Dakar Rally was to fit the driver and co-driver's doors, used in a serial production of SUVs, onto the space frame of the racing vehicle to be developed. The development of completely new doors, with high demands placed on water and dust proof sealing, only for this specific application would be too expensive and too complicated (Fig. 12.15).

One of the fields of the automotive industry where the RE methods can be widely used is the adaptation of serial production vehicles to the needs of persons with disabilities. In this case, the idea was to develop a rear door opening mechanism which would perform a translatory movement of the door, so that a person with disabilities could put their wheelchair more easily behind the driver's seat (Fig. 12.16).

The obtained 3D models represent a good basis for the design interventions and various analyses and simulation tasks. The use of optical measurement systems results in shorter overall development time.

12.6.2 Railway

Welded swivel-trucks (Fig. 12.17) are used in many European trains. Each swivel-truck needs to be measured and delivered with a measuring protocol. The measurement of these swivel-trucks, a task that a few years ago could have only been performed using tactile 3D CMMs or measuring arms, can be carried out easily and efficiently with a portable TRITROP photogrammetry system.

With TRITOP and an automated evaluation routine, one person is able to perform the measurement of such a swivel-truck and create a measurement report (Fig. 12.18) within forty minutes.

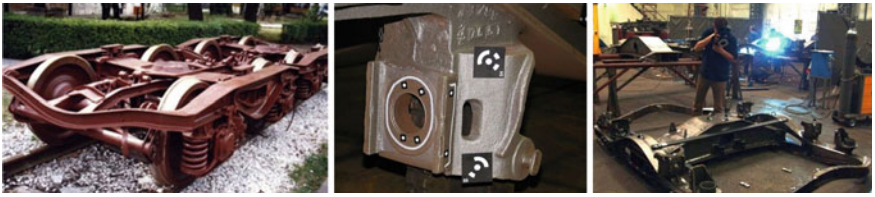


Fig. 12.17 Finished swivel-truck (*left*), a frame detail with the reference points and marked features (*middle*), the measuring process (*right*) [24]

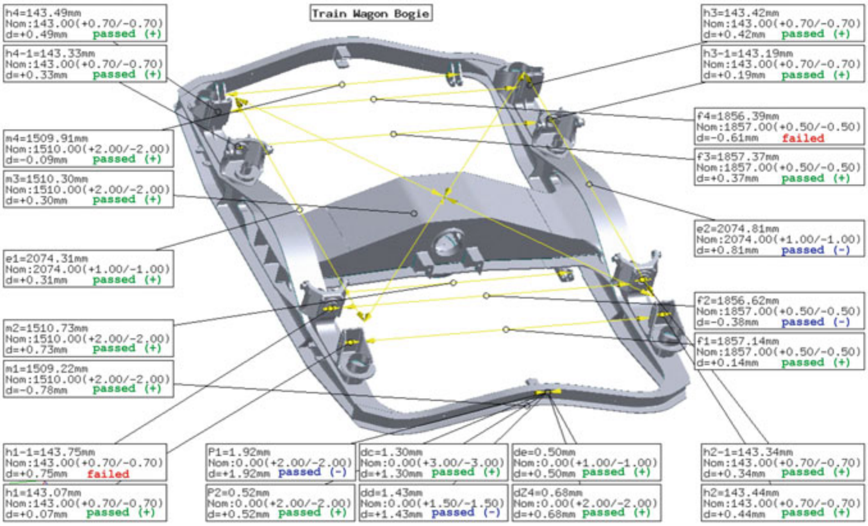


Fig. 12.18 Example of a measurement report. The deviations from the CAD data are displayed and evaluated (e.g. passed, failed, colour markings) [24]

According to [24], this time is required to apply the reference points, mark the characteristic features, place the scale bars, record about fifty images and transfer them to a laptop, carry out an automatic evaluation (the definition of the marker lines, the alignment of the measurement data with the nominal data, the calculation of the deviations) and to prepare and print out a measurement report.

12.6.3 Machinery

The case presented in [25] covers the measurement of a large iron casting for a wind turbine gearbox with an optical LED-based triangulation system (Fig. 12.19). As the castings undergo a final machining process, it is important to know the amount of excess material. The presence of sufficient excess material will be precisely

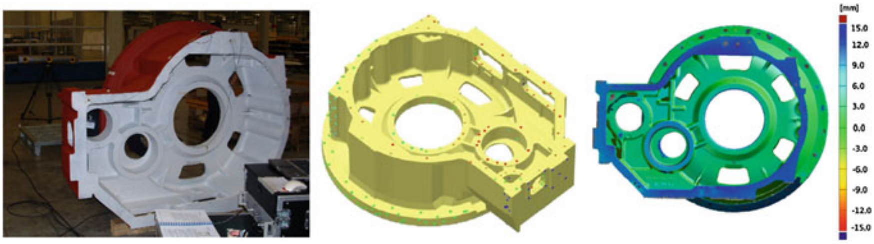


Fig. 12.19 Camera-based triangulation system in the back, the casting in the front (*left*), coloured dots indicating the deviation from the CAD file of the casting (*middle*), a colour plot of the casting scan compared with the CAD file of the machined part (*right*) [25]

determined by the alignment of the part on the machine. By measuring the castings before machining, it is possible to determine in advance the best suited alignment and detect the castings that deviate too much to fabricate good parts, so that at least the machining costs can be saved.

The results show that a systematic inspection of castings is an added value of the production process. By measuring the part, it is possible to determine the most suitable alignment for machining, so that the presence of sufficient excess material is guaranteed over the whole part. It is also possible to detect in advance bad castings, so that machining costs can be saved.

The RE methods are applied to various sizes of machine elements, from the production of replacement gaskets, presented in [26] to the retrofit of turbines presented in [27]. The main reasons for the turbine retrofit are extending turbine life time, improving reliability and operational flexibility, decreasing specific heat consumption and improvement of inner thermodynamic effectiveness. Typical steps which include RE methods in a retrofit process are shown in Fig. 12.20.

12.6.4 Architecture/Archaeology

In architecture, building reconstructions, city planning and similar projects a 3D data capturing of smaller scenes and large areas is needed. In such cases remote sensing is required. The most popular methods used are: static terrestrial laser scanning, terrestrial cinematic laser scanning from ground vehicles and airborne laser scanning from aircraft. All of them have their limitations. For projects that include a rapid and cost effective 3D data capturing of larger street sections, especially if they include tunnels (Fig. 12.21), terrestrial cinematic laser scanning could be the best solution.

Documentation of architectural and archaeological sites and monuments is an activity that requires the capturing of information from different sources. Experience has shown that it is possible to provide the necessary information with the required accuracy and completeness only by the integration of multisource data.

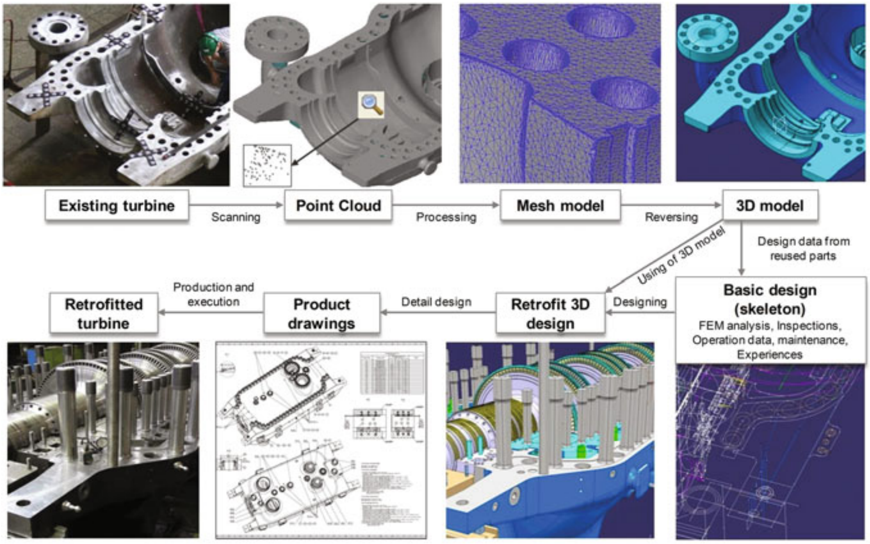


Fig. 12.20 Retrofit of a turbine [27]

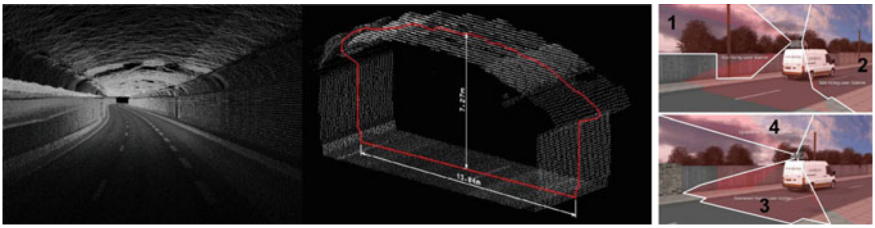


Fig. 12.21 A tunnel profile measured by the terrestrial cinematic 360° four laser scanning system StreetMapper [28]

A parallel use of geodetic and surveying measurements and photogrammetric data acquisition with imagery and terrestrial laser scans has proven to be an ideal combination, especially in large and complex monuments.

A successful 3D modelling of the Obelisk Tomb was achieved by integrating laser scanning technology with photogrammetry (Fig. 12.22). The approach combines the 3D models (developed from the range-based data) with multiple high resolution external images to yield photorealistic 3D models. The subpixel accuracy in the 2D-3D co-registration process between the images and the model is fundamental to matching accurately the texture from the imagery to the final 3D model without distortion. The presented approach is not only suitable for yielding high quality perspective views and photorealistic 3D models but it is also suitable for making amazing reality-based movies.

Fig. 12.22 View of the obelisk tomb and the Bab As-Siq triclinium in Petra/Jordan and a 3D view of the four-point clouds collected [29]

12.6.5 *Medical Applications*

Medical Reverse Engineering (MRE) is aimed at using the RE technology to reconstruct 3D models of the anatomical structures and biomedical objects for the design and manufacturing of medical products as well as Biomedical Engineering research and development. Different concepts and methodologies are provided to understand fundamentally the MRE processes and workflow. According to [7, 30], the key MRE applications are personalised implants for bone reconstruction, dental implants and simulations, surgical tools, medical training, vision science and optometry, orthopaedics, ergonomics, orthosis, prosthesis and tissue engineering. In addition, the RE methods are today indispensable in the Virtual Reality Surgical Planning (VRSP) process, whose flowchart is shown in Fig. 12.23.

12.6.5.1 Dentistry

In the case presented in [32], laser scanning was used to evaluate, by indirect methods, the accuracy of computer-designed surgical guides in the oral implant supported rehabilitation of partially or completely edentulous patients. Five implant

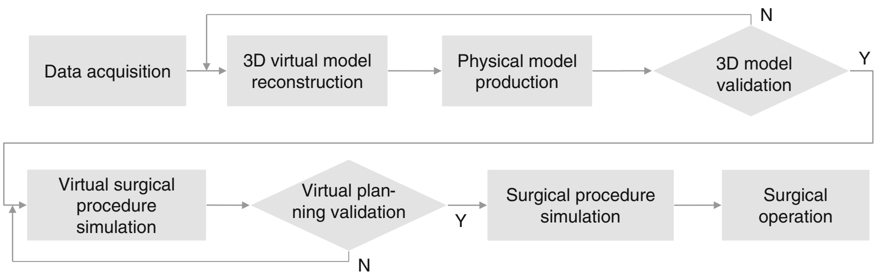


Fig. 12.23 Flowchart of the virtual reality surgical planning process [31]

supported rehabilitations for a total of twenty-three implants were carried out by computer-designed surgical guides, performed with the master model developed by muco-compressive and muco-static impressions. For all the cases, the VRSP process, starting from the 3D models obtained by the dental scan CT data, was performed. The implants were inserted in the pre-surgical casts in the positions defined in the process of virtual planning. These positions were acquired by three-dimensional optical laser scanning and compared with the laser scans of the intraoral impressions taken post-operatively. A comparison between the post-surgical implant replica positions and the positions in the pre-operative cast, made for five patients, shows the standard deviations within the range, which are absolutely negligible in the surgery (Fig. 12.24).

The results of this research demonstrate an accurate transfer of the implant replica position by virtual implant insertion into both a pre-operative and a post-operative cast, obtained from impressioning. In previous studies, the evaluation of the implant positions has required a post-surgical CT scan. With the indirect methods, using laser scanning techniques, this extra radiation exposure of the patient can be eliminated.

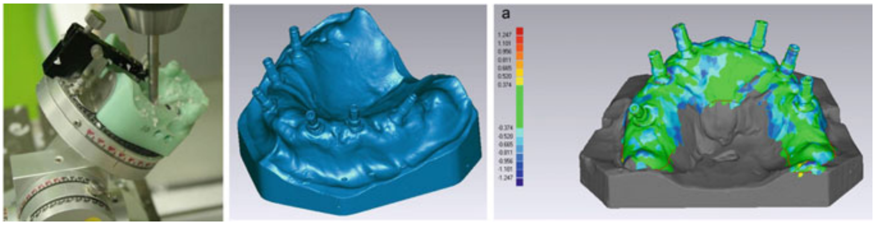


Fig. 12.24 The master model is oriented and drilled so that the implant analogues can be placed in it (*left*), a CAD model of the post-surgical cast (*middle*), CAD model comparison (*right*) [32]

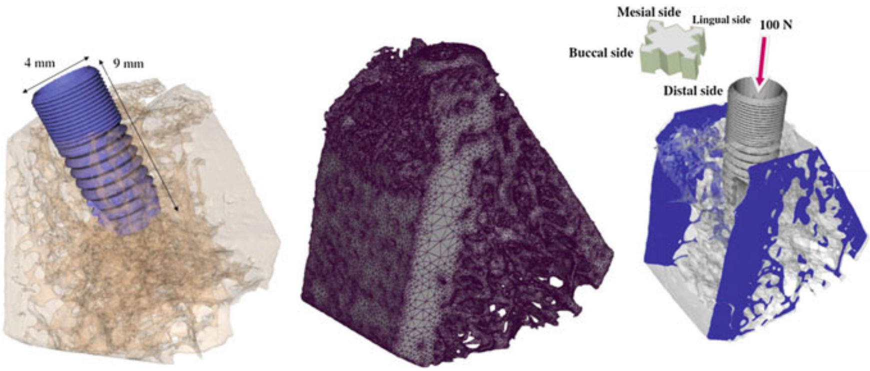


Fig. 12.25 Bone-implant complex (*left*), a 3D STL surface representation of the segmented μCT bone-implant complex (*middle*), the application of load (*red arrow*) to the FE model of the bone-implant complex and the enforcement of boundary conditions (*blue* surfaces) (*right*) [33]

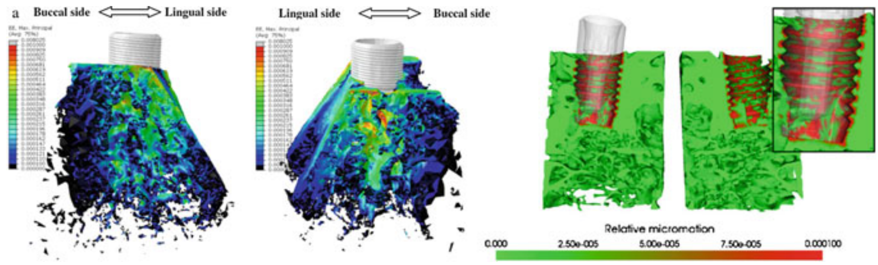


Fig. 12.26 Threshold plot showing the strain magnitude (equivalent micro strains) distribution within the bone micro architecture (*left*), an open view of the implant-bone complex showing the local displacements (micromotions of the bone with respect to the implant) of the trabecular architecture (*right*) [33]

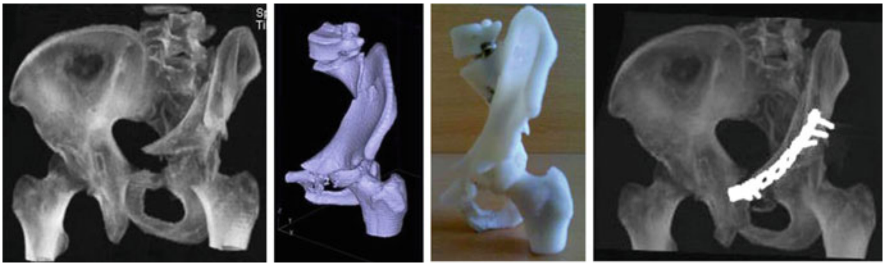


Fig. 12.27 Preoperative X-rays—a Judet obturator view (*left*), the stereolothographic image of the fractured acetabulum (*middle*, *left*), a rapid prototyping (RP) model of the fractured acetabulum (*middle*, *right*), a postoperative Judet view of the acetabulum (*right*) [34]

In the case presented in [33], the author's first objective was to assess the strain magnitude and distribution within the 3D trabecular bone structure around an osseointegrated dental implant loaded axially. The second objective was to investigate the relative micromotions between the implant and the surrounding bone. In order to reach these objectives, a μCT-based FE model of an oral implant implanted into a Berkshire pig mandible was developed along with a robust software methodology. The FE mesh of the 3D trabecular bone architecture was generated from the segmentation of μCT scans. The implant was meshed independently from its CAD file obtained from the manufacturer. The meshes of the implant and the bone sample were registered together in an integrated software environment (Fig. 12.25). A series of non-linear contact FE analyses, considering an axial load applied to the top of the implant in combination with three sets of mechanical properties for the trabecular bone tissue, was devised (Fig. 12.26).

The high level of resolution in the FE mesh of a novel μCT-based 3D FE model of the trabecular bone structure provided a new insight into the complex bone strain distribution pattern and showed that the calculated level of strain and micromotions in response to an axial load is in some qualitative and quantitative agreement with published experimental data, thus confirming the usefulness of μCT-based FE models in dental mechanics.

12.6.5.2 Surgery

The VRSP process is used in the cases of bone fracture with a complex geometry, presented in [34]. The production of a copy of the fracture or a deformity in a bone can be one of the important applications of the integration between two modern computer-based technologies, reverse engineering and RP (Fig. 12.27). This case presents the use of medical CT/MRI scanning, three-dimensional reconstruction, anatomical modelling, computer-aided design, RP and computer-aided implantation in treating a complex fracture of acetabulums, calcaneum and medial condyle of femur (Hoffa's fracture). This methodology reduces the surgical time, lowers the requirement of an anesthetic dosage and decreases the intraoperative blood loss.

12.6.5.3 Prosthesis Development

In [35] the authors proposed a new 3D design paradigm for the development of custom-fit soft sockets for lower limb prostheses. The new paradigm is centred on a digital model of the selected part of the human body and it is completely based on a computer-aided modelling and simulation of the two interfacing parts: the stump and the socket (Fig. 12.28). In such a context, different issues related to the human

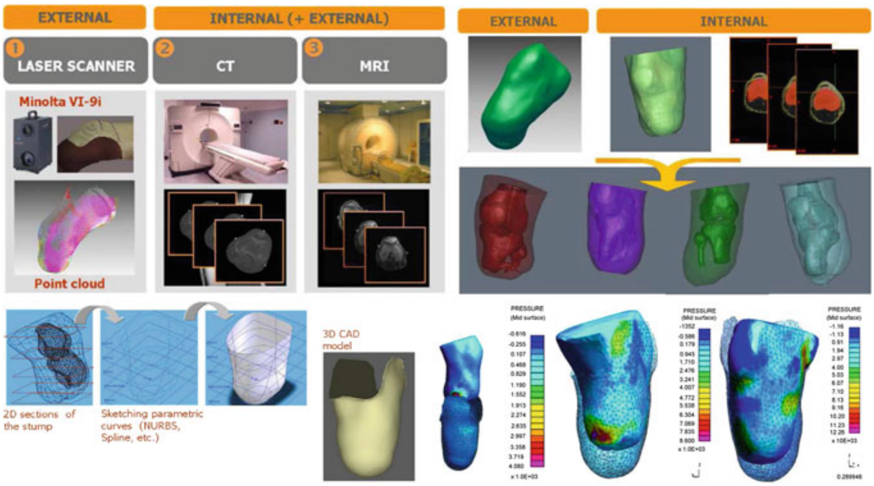


Fig. 12.28 Reverse Engineering equipment and techniques for morphology acquisition (*up, left*), digital models of the stumps of four amputees (*up*, *right*), the generation of the 3D model of the socket (*down*, *left*), the FEM simulation of the socket wearibility (*down*, *right*) [35]

body are considered: acquisition of the stump morphology, generation of a complete virtual model including both the external shape (skin) and geometry of the internal parts (muscles and bones) and mechanical characterisation of the stump, simulation of the socket-stump interaction, realisation of the physical prototype.

According to the authors, the proposed approach yields a better quality of the final product, a shorter involvement of the amputee implying a lower psychological impact, a limited use of physical prototypes and a shorter development time.

12.7 Ethical and Legal Issues

Intellectual Property Rights (IPRs) have become the key issue of the global innovation policy (see Chap. 18). They are generally protected by utility patents, design patents and copyright, but their strength varies from country to country. The Agreement on Trade Related Aspects of Intellectual Property Rights (TRIPS), signed in 1994 as a founding element of the World Trade Organisation (WTO), represents the most important attempt to establish a global harmonisation of the Intellectual Property protection. TRIPS, although an international treaty that obligates member states of the WTO to protect trade secrets, neither requires nor sanctions a reverse engineering privilege [36]. The World Intellectual Property Organisation (WIPO) assists developing countries in the implementation of TRIPS [37]. On the European level, the European Union Directive 2009/24/EC obligates Member States to protect computer programs by copyright, by analogy to the protection of literary works.

A standard legal definition of reverse engineering accepted by the U.S. Supreme Court (1974) is that it is a process of "starting with the known product and working backwards to divine the process which aided in its development or manufacture" [2]. The Supreme Court (1989) underlined the importance of reverse engineering, characterising it as an "essential part of innovation".

A prohibition on reverse engineering would seem to have two beneficial effects [36]: It increases incentives to introduce innovative products on the market, and it avoids wasteful expenditures on reverse engineering. However, reverse engineering has beneficial effects that must also be considered: it can create competition in the marketplace, leading to lower prices and it can spur second comers to introduce additional innovations into that market (Table 12.5).

Table 12.5 Social calculus of reverse engineering in manufacturing sector [36]

	RE legal	RE illegal
Incentives to innovate	Lower (but adequate)	Higher (but excessive)
Price	Lower	Higher
Follow-on innovation	Higher	Lower
Duplicated/wasted costs	Higher (but avoidable by licensing)	Lower

The role of RE is very well expressed by Samuelson and Scotchmer [36] who said that reverse engineering is fundamentally directed towards discovery and learning. Engineers learn about state-of-the-art engineering not just by reading printed publications, going to technical conferences and working on projects for their firms, but they also learn by reverse engineering others' products. Learning about what has been done before often leads to new products and advances in know-how. RE may be a slower and more expensive way of obtaining information to percolate through a technical community than patenting or publication, but it is nonetheless an effective source of information.

Analysing international activities in the field of IPRs over the last several decades, Archibugi and Filippetti [38] argue that the importance of TRIPS in the process of generation and diffusion of knowledge and innovation has been overestimated by both their supporters and their detractors. Although the main knowledge is today concentrated in the Western world, according to his opinion "TRIPS alone will not lead to an increase in the technology gap between western and emerging countries". Giving the summary of the current intellectual property global rule Henry and Stiglitz [39] claim that this rule may obstruct both innovation and dissemination and suggest reforms to foster the global dissemination of innovation and sustainable development.

12.8 Conclusions and Outlook

Engineering design is the process of devising a part (component), device, system, or process, focusing on engineering intuition, creativity and originality. On the other hand, reverse engineering is the process of discovering the technological principles of a part, device, system, or process through the analysis of their structure, function and operation. RE focuses on the recreation (reinvention) of the original parts, system, or process and includes alternative engineering solutions. In recent years, reverse engineering has become a standard practice for mechanical engineers who need to replicate or repair a worn part, or control quality of a produced part. Nowadays, reverse engineering has also become a practice often used by various experts in different areas. Consequently, demand for RE tools has become increasingly important and has led to the development of tools that are now commercially available. Data acquisition tools (hardware) and data processing techniques (software) are evolving rapidly. Tools and techniques are developing in terms of accuracy, acceleration of the process and use of RE methods in new applications.

In this chapter, basic information on RE methods has been presented. Particular emphasis has been placed on reviewing, classifying and comparing the most common RE methods and their applications in various fields of human activity. Further, a short review of some aspects of RE (data acquisition and processing, surface reconstruction, etc.) is included. The chapter also considers several important papers dealing with each of these aspects. The presented applications of RE highlight the wide range of tasks that can be solved by using RE methods.

So far, RE applications have been introduced in many areas, and typical applications include product development and manufacture (CAD-CAM-CAE), RP, quality control and inspection of mechanical parts, 3D graphics and animations, 3D art modelling (sculpturing), topography, architectural, archaeology and cultural heritage documentation applications, as well as biomechanical and medical applications. The applications also show that the RE methods are often used together with other methods which results in a multidisciplinary approach to problem solving. This approach requires the interaction and collaboration of various experts from different areas, including reverse engineering and RP, design and manufacturing, material sciences, biomedical engineering, medicine, etc. All experts, with different and complementary advantages and limitations, can improve RE activities, and such an approach is particularly effective when dealing with complex tasks.

Today, a wide range of tools for reverse engineering is available. It is often difficult to select the most suited tool or system for a specific task. All systems have their own particular strengths and weaknesses. When selecting a RE system, three main technical specifications should be kept in mind: task requirements, part restrictions and environmental restrictions. However, the use of data acquisition tools is not a trivial task: the use of the systems is still rather complex and skilled professionals are required to operate them. Also, data processing in most of the RE applications require high skills of image processing as well as design and geometrical modelling.

References

1. Chikofsky E, Cross J II (1990) Reverse engineering and design recovery: a taxonomy. IEEE Soft 7(1):13–17
2. Wang W (2011) Reverse engineering—technology of reinvention. CRC Press, Boca Raton. ISBN 13: 978-1-4398-0631-9
3. Pham DT, Hieu LC (2008) Reverse engineering–hardware and software. In: Raja V, Fernandes KJ (eds) Reverse engineering—an industrial perspective, Springer, London, pp 33–70
4. Raja V (2008) Introduction to reverse engineering. In: Raja V, Fernandes KJ (eds) Reverse engineering—an industrial perspective, Springer, London, pp 1–9
5. Raja V, Fernandes KJ (eds) (2008) Reverse engineering—an industrial perspective. Springer, London, ISBN 978-1-84628-855-5
6. Müller HA, Jahnke JH, Smith DB, Storey M-AD, Tilley SR, Wong K (2000) Reverse engineering: a roadmap. In: Proceedings on the future of software engineering, pp 47–60
7. Hieu LC, Sloten JV, Hung LT, Khanh L, Soe S, Zlatov N, Phuoc LT, Trung PD (2010) Medical reverse engineering applications and methods. In: 2nd international conference on innovations, recent trends and challenges in mechatronics, mechanical engineering and new high-tech products development MECAHITECH, 23–24 Sept 2010, Bucharest
8. Sansoni G, Trebeschi M, Docchio F (2009) State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine and criminal investigation. Sensors 9:568–601. doi:10.3390/s90100568

9. Barbero BR, Ureta ES (2011) Comparative study of different digitization techniques and their accuracy. Comput-Aided Des 43(2):188–206
10. Bi ZM, Wang L (2010) Advances in 3D data acquisition and processing for industrial applications. Robot Comput-Integr Manuf 26:403–413
11. Chang M-C, Fol Leymarie F, Kimia BB (2009) Surface reconstruction from point clouds by transforming the medial scaffold. Comput Vision Image Underst 113(11):1130–1146
12. Di Angelo L, Di Stefano P, Giaccara L (2011) A new mesh growing algorithm for fast surface reconstruction. Comput-Aided Des 43(6):639–650
13. Labatut P, Pons J-P, Keriven R (2009) Robust and efficient surface reconstruction from range data. Comput Graphics Forum 28(8):2275–2290
14. Ye X, Liu H, Chen L, Chen Z, Pan X, Zhang S (2008) Reverse innovative design—an integrated product design methodology. Comput-Aided Des 40(7):812–827
15. Wand M, Adams B, Ovsjanikov M, Berner A, Bokeloh M, Jenke P, Guibas L, Seidel H-P, Schilling A (2009) Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data. ACM Trans Graphics 28(2):15
16. Ihrke I, Kutulakos KN, Lensch HPA, Magnor M, Heidrich W (2008) Transparent and specular object reconstruction. Comput Graphics Forum 29(8):2400–2426
17. Rocchini C, Cignoni P, Montani C, Pingi P, Scopigno R (2001) A low cost scanner based on structured light. Computer graphics forum (Eurographics 2001 Conf. Proc.) 20(3):299–308
18. Page D, Koschan A, Abidi M (2008) Methodologies and techniques for reverse engineering—the potential for automation with 3-D laser scanners. In: Raja V, Fernandes KJ (eds) Reverse engineering—an industrial perspective, Springer, London, pp 11–32
19. Savarese S (2005) Shape reconstruction from shadows and reflections. PhD Thesis, California Institute of Technology
20. Post D, Han B (2008) Moiré interferometry. In: Sharpe WN Jr. (ed) Springer handbook of experimental solid mechanics. Springer, Berlin, pp 627–645 (ISBN: 978-0-387-26883-5)
21. Magnet lab (2014) http://www.magnet.fsu.edu/education/tutorials/magnetacademy/mri/. Accessed 08 Mar 2014
22. Zhang YF, Wong YS, Loh HT (2008) Relationship between reverse engineering and rapid prototyping. In: Raja V, Fernandes KJ (eds) Reverse engineering—an industrial perspective, Springer, London, pp 119–139
23. Lulić Z, Tomić R, Ilinčić P, Šagi G, Mahalec I (2012) Application of reverse engineering techniques in vehicle modifications. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering (2013), vol 2. Trier, Springer, ISBN 978-1-4471-4425-0, pp 921–932
24. GOM (2008) Application example: quality control, sheet metal: measuring characteristic features using the optical measuring machine TRITOPCMM. GOM mbH, Rev. A (en) 03042008, http://www.gom.com/fileadmin/user_upload/industries/tritop_cmm_EN.pdf
25. Cuypers W, Van Gestel N, Voet A, Kruth J-P, Mingneau J, Bleys P (2009) Optical measurement techniques for mobile and large-scale dimensional metrology. Opt Lasers Eng 47 (3):292–300
26. Tut V, Tulcan A, Cosma C, Serban I (2010) Application of CAD/CAM/FEA, reverse engineering and rapid prototyping in manufacturing industry. Int J Mech 4(4):79–86
27. Červenková L, Skovajsa M (2013) Doosan škoda power: modernization and retrofitting of turbine components. GOM conference—optical metrology 2013, 9–12 Sept 2013, Braunschweig, Germany
28. Kremer J, Hunter G (2007) Performance of the StreetMapper mobile LiDAR mapping system in "real world" projects. In: Fritsch D (ed) Photogrammetric Week'07, Wichmann, Heidelberg, pp 215–225
29. Lerma JL, Navarro S, Cabrelles M, Seguí AE, Haddad N, Akasheh T (2011) Integration of laser scanning and imagery for photorealistic 3D architectural documentation. In: Wang C-C (ed) Laser scanning, theory and applications. InTech, Rijeka, ISBN: 978-953-307-205-0

30. Hieu LC, Zlatov N, Sloten JV, Bohez E, Khanh L, Binh PH, Oris P, Toshev Y (2005) Medical rapid prototyping applications and methods. Assembly Autom 25(4):284–292
31. Robiony M, Salvo I, Costa F, Zerman N, Bazzocchi M, Toso F, Bandera C, Filippi S, Felice M, Politi M (2007) Virtual reality surgical planning for maxillofacial distraction osteogenesis: the role of reverse engineering rapid prototyping and cooperative work. J Oral Maxillofac Surg 65(6):1198–1208
32. Giordano M, Ausiello P, Martorelli M (2012) Accuracy evaluation of surgical guides in implant dentistry by non-contact reverse engineering techniques. Dent Mater 28(9):178–185
33. Limbert G, van Lierde C, Muraru OL, Walboomers XF, Frank M, Hansson S, Middleton J, Jaecques S (2010) Trabecular bone strains around a dental implant and associated micromotions—a micro-CT-based three-dimensional finite element study. J Biomech 43 (7):1251–1261
34. Bagaria V, Deshpande S, Rasalkar DD, Kuthe A, Paunipagar BK (2011) Use of rapid prototyping and three-dimensional reconstruction modeling in the management of complex fractures. Eur J Radiol 80(3):814–820
35. Colombo G, Filippi S, Rizzi C, Rotini F (2010) A new design paradigm for the development of custom-fit soft sockets for lower limb prostheses. Comput Ind 61(6):513–523
36. Samuelson P, Scotchmer S (2002) The law and economics of reverse engineering. Yale Law J 111(7):1575–1663
37. WIPO intellectual property handbook (2004) WIPO, 2nd edn. ISBN 978-92-805-1291-5, reprinted 2008
38. Archibugi D, Filippetti A (2010) The globalization of intellectual property rights: four learned lessons and four thesis. Glob Policy 1(2):137–149
39. Henry C, Stiglitz JE (2010) Intellectual property, dissemination of innovation and sustainable development. Glob Policy 1(3):237–251

Chapter 13
Digital Mock-up

Roberto Riascos, Laurent Levy, Josip Stjepandić and Arnulf Fröhlich

Abstract Product development in the mobility industry is characterized by extreme time-to-market, high product complexity, cost pressure and many geographically dispersed stakeholders. Thus, efficient control mechanisms are necessary to manage a seemingly unmanageable project successfully and to achieve a strong finish. Digital mock-up (DMU) serves, as a central validation instrument in such a complex scenario, not only to visualize spatially the current status of the virtual product but also to evaluate the project's progress. In conjunction with a high-variant product structure, as it is the case in modern vehicles, the use of DMU makes the check of the spatial consistency of the overall product possible, taking over what today's CAD and PDM systems alone are not capable of. Taking the function of the product into account, the result is the so-called functional DMU (FDMU) which aims at facilitating the direct experience of functions on the virtual model in the overall context of the product. While DMU offers a visual straightforward human interface for control, DMU creation, calculation and processes can be automated well, so that the spatial test (collision check, assembly check) can be performed for all conceivable product variants in batch during the night). Nevertheless, human intervention is still required for the solution of design conflicts. Although all current problems are not yet solved in the context of DMU, leading PLM vendors do offer powerful tools to support the DMU process. Due to its central role in the development process DMU is subject of intensive research and development for speeding up the process and to increase accuracy.

R. Riascos (✉) · J. Stjepandić · A. Fröhlich
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: roberto.riascos@googlemail.com

J. Stjepandić
e-mail: josip.stjepandic@prostep.com

A. Fröhlich
e-mail: arnulf.froehlich@prostep.com

L. Levy
Airbus SAS, Toulouse, France
e-mail: laurent.levy@airbus.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_13

Keywords Digital mock-up · DMU · CAD · VR · Product structure · Product validation

13.1 Introduction

While for a long time the term mock-up denoted a preferably realistic replica (mostly made of cheap material) of a complex material product for illustrative or installation purposes, with the entry of CAD and efficient computer graphics, the digital mock-up (DMU) accrued. The term DMU indicates a computer generated, true-to-life product model, successively built up within a development project following the project progress, to replace a part of the very expensive, real product-testing by computer simulations. The idea of DMU arose from the need for replacement of the cost-intensive and time-consuming physical test carriers (physical mock-up) by means of computer support. Objectives of the DMU are the replacement of as many physical mock-ups as possible and the provision of various, up to date and consistent views of a product's shape and function in the form of virtual models.

The DMU is one of the parts of product lifecycle management (PLM). Considering PLM as a holistic approach for the management of products from initial requirement to decomposition, DMU is the PLM component displaying the product at any point of the Product Lifecycle including all development stages and variant diversities. Based on a DMU, a plurality of validations such as installation and removal tests, interference checks and build-ability checks can be performed. Therefore, the term *Digital Engineering Validation* is used as a synonym for DMU in many companies too. Furthermore, DMU can be understood not only as a technique, but as a superordinated cross-domain engineering process with aim to continuously provide a transparent view on the progress of a development project by using workflows defined in product data management (PDM) systems and various visualization techniques and tools.

Depending on the application, DMU can be used company-internal, or even in collaboration scenarios with external development partners [1]. Thus, DMU represents an important methodology in distributed, global product development processes.

The structure of this chapter reflects this aim. In the following Sect. 13.2, the historical background of DMU and the need for DMU in particular in context of concurrent engineering (CE) are briefly introduced. The differences between CAD and DMU are also outlined. All components which are needed to fully build and maintain a DMU are explained in Sect. 13.3. In the following Sect. 13.4 functional capabilities of an efficient DMU are described: communication, analysis, management, completeness, and process automation. Subsequently, practical commercial applications are shown in Sect. 13.5, classified as standard and customized solutions. Recent research activities have sought to introduce functional validation

[Functional DMU (FDMU)] (Sect. 13.6). Subsequent Sect. 13.7 showcases the practical achievements by three typical DMU applications for the aerospace and automotive industry. A conclusion section gives insight into the future of DMU from a CE perspective.

13.2 Evolution of DMU

DMU is a re-encounter of the original and intuitive design process with its need to have a reliable representation of the product that is easily assimilated not only by the designer but other actors involved in the product's life cycle with different technical education and needs. Primordial lifecycle process consisted in the variation of time consuming and expensive prototypes or physical mock-ups (hereon called merely mock-up). The lack of a widespread set of rules and agreements to realize an organized and documented mock-up based product conception flowed into the introduction of a series of symbols and conventions illustrated in 2D technical drawings whilst winning an unambiguous interpretation of the depicted information but, at the same time, paying the price of losing one dimension and therefore understandability for a broader stakeholder audience. With the development of digital technologies and the arrival of systems for 2D technical drawings, more actors could be again involved in a concurrent design. Technical drawings still did not fulfill the need for mock-ups, although their interpretation among expert groups delivered sufficient information.

In the 1980s, CAD software editors went progressively to 3D representation, at first in a wireframe representation and later as solid. The step from 2D technical drawings to 3D models closes the lost dimension gap and opens the possibility to produce a digitally created mock-up regaining most of the mock-up's virtues and surpassing them with a relative low cost creation.

A further step in the direction of a DMU was taken when companies like ComputerVision with CAMU, Dassault Systèmes with CATIA Session and VPM, understood that representing the product as an assembly was the step beyond to represent a complete product. They used the bill of material principle, already valid (and still) in all industry, to adapt their product structure software to the need. In the 90s, some industrial companies added their configuration management system into the DMU to manage the evolution of their products.

DMU evolved further while the CAx chain broadened. DMUs overtook several test scenarios of its predecessor, lowing costs, booting time to market and increasing quality of the product, while, allowing a higher product complexity.

Mock-ups are still widely used, especially in the manufacturing phases of the product life-cycle where they prove the fulfillment of expected requirements in geometrical design. In the further phases prototypes are used, being among them the difference that prototypes fulfill some kind of functionality of the real product, whereas the mock-up just geometrically represents the product. In this order of ideas there is also the difference between a DMU and a Virtual Prototype. The DMU reduces itself to the mere product presentation, leaving functionalities aside. The functionality extensions of a DMU will be discussed in Sect. 13.6. A further extension of the DMU to give the designer the possibility to virtually have a feel of the product or virtually touch the mock-up to improve its operability is reached combining a virtual and physical model in a hybrid [2] approach. This approach is especially useful in the design of household apparatus and cell phones where the user interface has a very high priority.

Mock-ups themselves have also lived a parallel digital evolution. Instead of being digitalized their creation is being updated to the modern technologies. Mock-ups were usually made of an alternative material like wood an ever rising tendency goes to create them in other materials like plastic, using stereo lithography and 3D printing.

13.2.1 A Need for Digital Mock-up

Reacting to today's market requirements companies have high pressure to keep constantly lowering times and costs with raising quality and innovation [3]. Models are often renewed as a reaction to the market. E.g. in the 1980s a car's model was renewed every 10 years, whereas this nowadays happens every 4 years. This major constraint pushed manufacturers to invest on CAD technologies in order to adjust to the market's time, quality and innovation needs. These investments plus others in digital technologies allowed companies ever more to virtualize several phases of the product life cycle especially in the development phase. Giving the starting point of a CE strategy crossing several departments and actors united around the same information basis. Engineering designers, stress specialists, producers were able to work together, nearly at the same time, to create the better compromise for the product.

Although a high investment is needed to create a DMU, the cost reduction effect of using a DMU as a central platform can be objectively [4] and economically measured and is noticeable in other fields like conceptual layout, communication, decision taking, CE, prototype creation, maintenance, retrofit engineering and product recycling among others.

Industries are more and more in a worldwide competition, where alliances with international partners or suppliers are becoming important for enlarging their markets, to find cheap partners, and to ensure political agreements [5]. Therefore engineering and manufacturing must communicate fast all around the world. Network bandwidth is in some cases too limited to exchange live data. Nevertheless, data exchanges with certain frequencies are sufficient to work efficiently with far laying partners or suppliers.

13.2.2 *Digital Mock-up in Concurrent Engineering*

In order to allow several designers from different disciplines to work on the same product, in the same spatial entities, at the same time, a CE strategy using a DMU as the central axis of information exchange is mandatory. A DMU has to be supported by master PDM systems that manage the complementary information to the geometry.

Managing information in a central or master PDM system from which a DMU is created enables to create information in a visual and fast representation easy to assimilate reducing redundancy in information management.

13.3 Components of a Digital Mock-up

A DMU consists of three types of product data: product structure, geometrical data, and meta data. The product structure is a hierarchical decomposition of the product into nodes, from the top node, the product itself, in main and sub components. This description dismembers the product in spatial or functional nodes that at the same time get dismembered until reaching the last possible entity, namely the geometrical data. High end and modern CAD systems allow at the same time an internal product structure within the geometrical description, this information gets lost while creating a DMU raising the issue of a black-box product structure, which can be both source of problems and solutions, while handling the DMU. In the product structure may sometimes contain the position of each element in space or relative to its parent node in the structure. The position is a 3D matrix with the space translation in the x, y and z coordinates plus the rotation of the element in space relative to the x, y and z coordinates. This information is at best managed in a PLM system using a single database.

The geometrical data is derived from the 3D CAD models that describe each element of the product in form and function. This description is usually simplified, thus losing information from the original data, like features, design history, color and texture among others. Simplified data, although possessing lesser information, has the advantages of being lightweight and facilitating automation processes. A simplified, tessellated 3D representation of a product, which is no more than a collection of neighboring triangles that wrap the original geometry giving a fairly good description of it, allows an automated space and collision analysis treating the geometry as collection of finite elements using known algorithms of this discipline.

All other information that describe the product like version, naming, numbering, description, material, product manufacturing information (PMI) among several engineering parameters, which are usually managed in a PDM system, complete the basic information needed to build a DMU. This information can be complemented with the use of extra data from other coupled systems.

The combination of DMU data with real life imagery captured in real time, live with video cameras produce an augmented reality (AR) or mixed reality (MR). AR/MR tools replace computer algorithms for geometry positioning, e.g. in assembly planning, where instead of hardcoding the movement of geometrical data, the user can move the geometrical object within the data imported from the cameras and then save it [6].

13.3.1 Building a Digital Mock-up

There are several ways to describe a product through its component decomposition in the form of a product structure. In order to build a DMU there are several approaches to do this product structure:

1. Avoid loops in the product structure
2. A DMU product structure is rather flat (horizontal) than long (vertical)
3. In the decomposition of the product prefer a space based before a functional based breakup.

The main goal of a DMU is to achieve a useful visualization of the product. In big and complex products a DMU is needed to analyze spaces where several assemblies for different functions coexist. For this reason is recommendable to prefer a space decomposition of the product before a functional decomposition where assemblies that not necessarily have a functional relationship with each other will have a common parent node.

While choosing the right product structure of a product the node decomposition should aim to have a broad, flat, horizontal product structure with as many assemblies as possible parallel to each other. Achieving all assemblies and geometry at the same product structure depth is impossible. Therefore, it is necessary to define a boundary where all nodes at the same level take the same space or function and fulfill only an administrative role. Below this boundary is ideal to place a configuration level where the high level assemblies below can be steered into the different product variations. Above the configuration level, the product structure will be called upper level, below lower level. In the lower level, node decomposition can be done freely as needed. For the decomposition of a lower level for a DMU is recommended not to mix assembly elements with parts with geometry in the same assembly. This problem is easily avoided through the insertion of an extra node under the assembly node and the geometry (Fig. 13.1).

A resulting product structure without loops arises, because all high level assemblies should be at the same level.

The necessary metadata to create a DMU is all related to name and numbering of elements, variant and configuration data that create the link between product structure and geometry and are necessary to build a bill of materials (BOM). Further than the necessary data to create a DMU, all other data that wants to be visualized must be related to the metadata e.g. maturity, origin (own, sub-contracted, bought, etc.), positions of moving parts, etc.

Building a DMU is a three layered process: creation of product data, management and integration of product data, and visualization of product data.

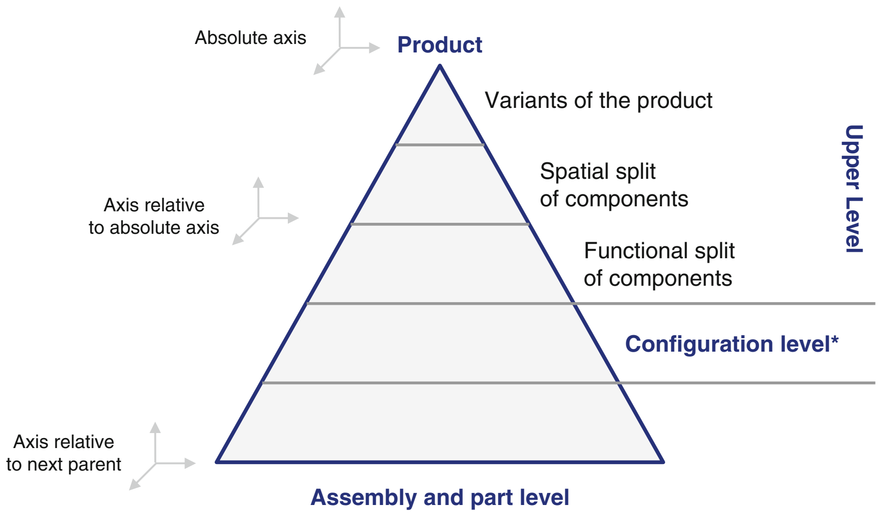


Fig. 13.1 Product structure decomposition

13.3.1.1 Creation of Product Data

In this layer of the process all product data that will be visualized in the DMU is created. This indicates the production of geometrical data but also all other data that describe the product, most of which is based in the geometrical description and the requirements of each element or part.

Geometrical data is created with 3D CAD software resulting in a 3D description of the product and its parts. Working on the basis of a product structure, each leaf of this structure is an element to be modeled (or directly created from a laser scan session usually done in retro-fit activties). Elements that are geometrically, spatially or functionally bound together form an assembly, assemblies can be bound together with other assemblies, the sum of all assemblies and elements that are not related to any assembly is the geometrical product description.

Modeling for a DMU

Modeling in modern 3D CAD depends on the capabilities of the 3D CAD system, the geometry elements available for the modeling process (points, lines, planes, surfaces or volume elements) and the internal mathematical description of these elements (analytical or parametrical). These capabilities are especially important at the point of the data conversion. The visualized data in the DMU must not always be the original CAD data, in the process of converting CAD data into lightweight visualization data there is a chance of failure in data conversion due to a wrong modeling strategy or conversion algorithm. Modern CAD systems offer the use of all types of geometrical elements and their combination with a parametrical description.

In some cases, the details can cause data conversion problems, especially with the geometry of the part which has too many triangles while converting. These failures are common around corners where roundings meet, threads etc. In these cases a less detailed or intentionally simplified model of the part is better for generating a good DMU. The user has to think, if in the case of the thread of a screw for example (not meaning a spiral conveyor) is necessary for DMU purposes or further work in the CAx chain (Fig. 13.2).

The granularity of a part within an assembly is a decision of the designer, taking into account the reusability of the parts. If two parts are always used together, a mini assembly with the two parts is recommended. An approach to mature the geometry in the DMU is to use a three phase plan to reach the fullest possible maturity for a DMU useful for the CAx chain.

In the first step the part or geometry is just represented with a basic volume (box, cone, pyramid, sphere, torus, etc.) containing also in simplified form the interfaces to other parts. To represent this interface, usually, a cone or pyramid is used to define additionally the direction of the interface. This first model fulfills the high level requirements to the part or assembly like space volume, position and interfaces (Fig. 13.3).

In the second phase the form of the part is recognizable, although, certain details like rounding, tapers and threads are left out. In this phase, it is possible to take the decision to split the part into several parts and, thus, create an assembly. This model fulfills the next level of requirements regarding form and function. Models in this phase are suitable for further analysis using the DMU (Fig. 13.4).

A third and last phase defines in every detail the form and function of the part including interfaces and relationships to interfacing parts or systems. This model has every necessary detail and fulfills the rest requirements for the part like weight, center of mass, etc. (Fig. 13.4).

FILE FORMAT	FILE SIZE [KB]	
.prt | Native CAD	432	208
.jt | Tessalation of the file above	156	16

Fig. 13.2 File size of a screw with and without the thread feature in native CAD and JT

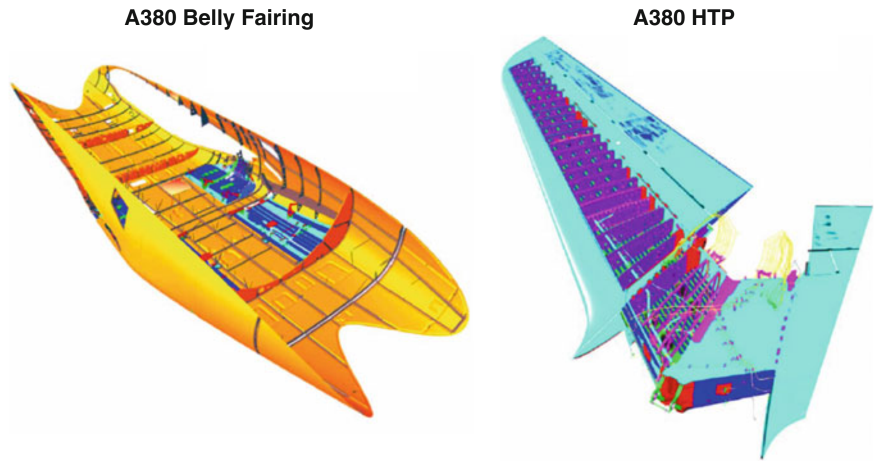


Fig. 13.3 Initial phase of DMU evolution, space allocation mockups [7]

Fig. 13.4 Example of the three phases of DMU evolution

Product Structure for a DMU

There are three strategies for 3D modeling and assembly building: bottom-up modeling, top-down modeling and a combination of both modeling strategies. CAD Systems offer the possibility to create an internal product structure inside the geometrical file or in a separate file that contains the product structure information. This functionality can be a disadvantage to create a DMU.

Assemblies are built using the constraints among geometrical data, e.g. a bolt fits concentrically into a hole as far as the bolt end fits the hole's bottom. This relationship is saved within the CAD system as a relationship between geometrical attributes of the CAD parts; the axis of the bolt goes through the center of the hole's bottom that at the same time touches the face end of the bolt.

Several problems emerge in this approach if the bolt and the hole do not belong to the same design team. The team responsible for the hole may change the hole's deepness, the bolt would lie deeper into the hole and thus all other geometrical dependencies to the bolt would move, probably deforming appearance of the DMU as a whole. This may also be the case when a part within the assembly is replaced

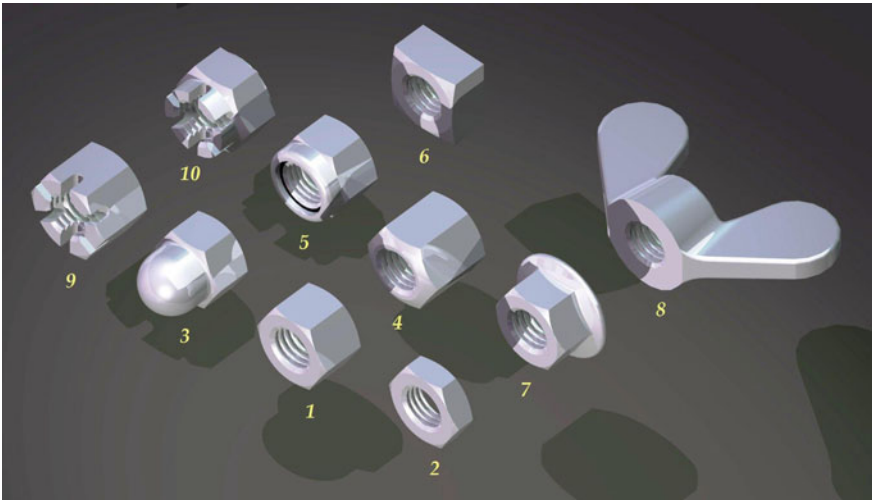


Fig. 13.5 Variants of the same product

by a variant of itself (Fig. 13.5). While these relationships are useful, for the DMU it is better if the geometry is placed relative in space to the origin of the parent node, avoiding changes that occur in the dynamical nature of CE [8].

In order to perform further analysis based in the DMU like completeness and maturity of the product a differentiation between external parts and assemblies has to be done in the DMU. Analogous to the dilemma between make and buy, where made parts are repaired and bought parts are replaced, in the DMU parts or assemblies that are made are versioned and those that are bought are replaced (mostly with a data exchange procedure depending on the level of integration of the supplier in the PDM system).

Management of Product Data

The ideal software to manage product data are PDM systems (see Chap. 16). In these systems and important for the creation of a DMU is the management of the product structure through which the geometrical data is managed. PDM systems can not only manage the huge amount of product data created during the lifecycle process but ensure a correct and secure data access to the right departments, teams or people in the correct format depending on predefined work-flows that are also managed within the PDM system.

Due to the increasing demand for a stronger and more efficient cooperation, proprietary systems are either replaced or web-based PDM systems are established, in which people with incompatible applications or operating systems meet on the internet and communicate with each other. Here, documents can be viewed and edited together.

PDM systems are supported by configurable computer-systems, so that every user can only view information and documents relevant to her/him and has no access to the remaining documents.

13.3.2 Maintaining a Digital Mock-up

Maintaining a DMU in completely updated state and available to all parties requires a support team that has the professional understanding of the processes around a DMU. Another strong aspect to take into account is the security; all the positive aspects of a DMU make it prone to be a security problem.

Summarized, maintaining a DMU is divided in three essential aspects: infrastructure, security and services. While the infrastructure and security aspects may fall into a wider IT strategy of a company, the services needed for the DMU are specific and need specially trained personnel to be accomplished.

13.3.2.1 Infrastructure

Working with a DMU obliges enterprises to do an investment in IT infrastructure split in network, equipment, hardware and operational sites. High speed and broadband networks are needed to exchange the necessary amount of data to guarantee an updated status of the DMU. These networks must reach all sites involved in the life-cycle process using DMU data. Working with PDM and CAD software obliges enterprises to acquire high performance computers that can cope with the computing strain imposed from data base queries and heavy weight computer graphics.

To ensure a correct and thrift use of the DMU these infrastructural aspects need to be maintained, kept up to date and continuously supported.

13.3.2.2 Security

Whilst the DMU contains a complete description of the product, securing the data contained in the DMU is a primary issue. The DMU itself as a light weight digital product description is prone to be stolen. The fact of insuring a well spread DMU for all necessary recipients is its biggest weakness, only controllable through thorough on-site security. The outlay of the IT infrastructure must take all security issues into account. Off-site suppliers must be contractually liable for the security of the exported data and data exchange processes must be encrypted and secured against data theft.

13.3.2.3 Services

The following DMU aspects have to be met through services:

Complete: a DMU is only useful if all the parts within a space that is to be analyzed are loaded. E.g. a moving part analysis is worthless when the whole environment is not loaded and a moving collision with all parts is not taken into account. In the definition of completeness is also a full data conversion meant. All CAD data has to be successfully and faithfully converted to the DMU file type.

Up-to-date: the geometrical and product structure information of the DMU has to have the latest data in order to be useful. This is the case even in the first design phases, where the detailed data is not visualized but rather the pre-design models. All data from suppliers that have no direct management of their data in the product structure master to the DMU needs to be updated. All metadata from third systems that want to be integrated in the DMU and visualized have to be updated.

Available: the DMU hast to be available to all needed instances to guarantee a flawless CE process. All access the third party systems with data what needs to be visualized has to be granted. Dedicated DMU for certain focus groups e.g. boxed, configured, functional, reduced DMU has to be available.

Data exchange: although the supplier must be contractually made responsible for a correct data delivery, a service of data exchange has to be offered to linked suppliers. These must be supplied at first with a valid and specific number range which they must so that data is problem free integrated into the PDM system. Suppliers also need check quality standards and tools defined by the manufacturer, among these count CAD geometry checkers, PDM checkers, tools to evaluate the tesselated or facetted quality of the data, etc.

13.4 Capabilities of a Digital Mock-up

In this section main functionalities and application areas of DMU are described.

13.4.1 What You Can Achieve with a Digital Mock-up

The use of a DMU allows the transfer of competences and responsibilities to external and offshore specialized partners, where local as well as external partners can work in neighboring parts of the same working space at the same time. This results in shortening the time to market while at the same time increasing the quality with the integration of specialized stakeholders while keeping the benefits of reducing costs with expensive mock-ups.

A DMU creates a virtual environment where complex products can be efficiently managed in all phases of the life-cycle [9] supporting simulations, reducing costs and risks delivering high reliable results [10].

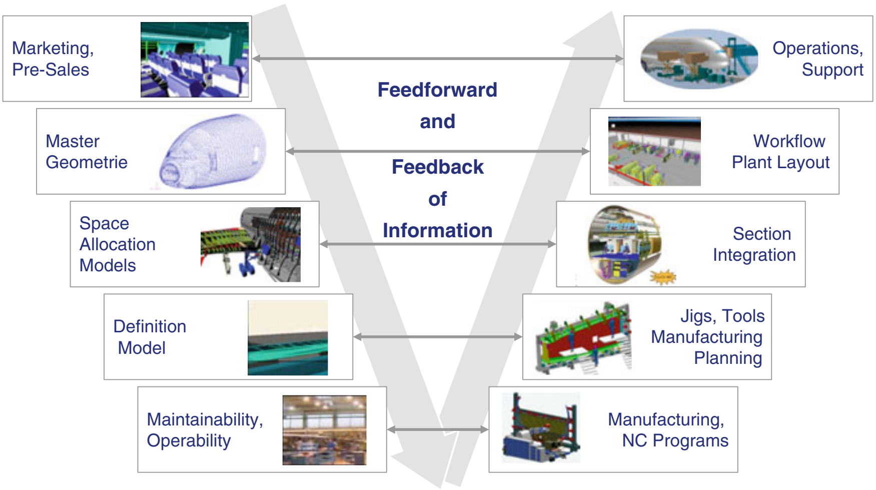


Fig. 13.6 Exemplary use of DMU in the aerospace product creation process

- Human simulations such as the RAMSIS (Computer Aided Anthropometric Mathematical System for Passenger Simulation) human model used in the automobile industry or simulation of human movement to simulate ergonomic cases [11].
- Simulation of environments, where physical variables such as gravity, temperature, lighting, stress and relative movement are digitally changed.
- Virtual machines with virtual machining and manufacturing [12].
- Product behavior simulations [13].

An end-to-end use of the DMU in the phases of life-cycle of the product is shown in Fig. 13.6.

13.4.2 Communication

Design reviews evaluate the technical progress toward meeting specification requirements. In these reviews multiple stakeholders with sometimes opposite agendas get together to decide the approval or rejection of technical solutions.

Using a DMU enhances the efficiency and development of reviews starting in a very early stage the verification process [14]. Enhancing the use of DMU with advanced telecommunication systems and media such as teleconferencing, common environments and virtual rooms allows stakeholders to be part of the review without having to be physically present in the same room, thus allowing all necessary stakeholders be part of the event. A common checklist and protocol of the review also leave a history of the methodical and structured process that derived into a decision, which in itself can then be as metadata added to the information available from the DMU [15] enriching the life cycle documentation and collaborative work [16] with different levels of interaction. Important in this case is thorough documentation of the reviews to recall and track decision making.

Decision making has to be bound to design rules based on clear requirements. Reviews are then necessary when a deviation of the requirements is proposed or when requirements of different systems collide or conflict with each other. In most cases when a compromise is needed, the DMU as a highly understandable product description enables the decision making of all stake holders in the levels of coordination, cooperation and collaboration (see Chap. 7) [1].

13.4.3 Analysis

A DMU allows users to perform digital analysis on the product representation without needing to destroy or damage the mock-up. Performing analysis of several types on the DMU is the most common use of it besides mere visualization. The most performed analysis are static collision and clearance detections which serve as starting point for analysis of further complexity such as dynamic collision detection, assembly and use analysis etc.

13.4.3.1 Collision and Clearance Analysis

Collision analysis aims to find allocation errors in the DMU before they have to be corrected, sometimes with use of brute force, in the mounting site. Collision detection is a highly automatable DMU process; almost all vendors offer a collision or clash detection module. Mathematically a collision detection is the analysis of the intersection of two planes spanned inside the boundaries of two triangles, where each triangle is part of the tessellated representation of the parts analyzed. These calculations fall under the realm of computational geometry.

The analysis starts with loading the DMU consisting of parts and the surrounding environment and then continues with calculation of the geometry intersections in a static mode or if the movement of parts is simulated, in a dynamic mode taking into account technological, functional and space constraints.

The result of the analysis is usually saved and documented in a log file containing the name and the position of the two parts involved in the collision, details of the collision in the form of a polyline, details of the collision based in the volume, deepness and center of mass of the intersection. These results need to be reviewed by a team of experts who decide the relevance of each collision and, thus, start a redesign or change process.

Nevertheless, not all collisions need to be checked. Know-how must be used to filter found collision. Experience shows some types of collision can be ignored and do not need to be reviewed or at least where collisions must be expected or even intentioned. Some examples of these cases are:

- Collisions of screw threads with nut threads,
- Collisions against flexible parts such as isolation, foam, springs, seals, etc.,
- Collisions with parts with a press or tight fit,

Furthermore, filtering out known and duplicated collisions spares time in the review process [17]. Reviewing engineers must know the context of the design and collision in order to evaluate it. This analysis can be done in static or also taking into account dynamics and movement of bodies in the DMU.

13.4.3.2 Assembly Investigation

Using DMU data that represents with high fidelity the real product, engineers are able to digitally plan the assembly of a product and avoid expensive changes if errors in the assembly are found after production.

The use of DMU goes from conceptual assembly planning in very early stages of the product conception where product variations can be discarded because assembly processes and procedures cannot be fulfilled [18] or interactively with the user recording hand position and gripping movements in an AR environment combining DMU with real time camera videos and so defining the assembly sequence and procedures defining the assembly cycle.

13.4.4 Management

Tracking the DMU is a useful indicator to take management decisions. Completeness, quality, maturity and DMU milestones among other can help management stakeholders to specify internal strategies or external regarding clients, subcontractors and suppliers.

The high amount of DMU issues, for example, collisions or low completeness in a specific design team, could reflect a lack of experience in the design team, faulty communication with other stakeholders, underestimation of the design complexity or merely under-staffing resulting in delays in the project calculations and costs.

The DMU also gives management stakeholders the chance to understandably communicate with the client, and receive direct and accurate feedback that can directly flow into other stakeholders that must take action. It increases dramatically the transparency of project progress and, therefore, helps identify any issue in an early phase.

13.4.5 Completeness

The design completeness of a product can be calculated based on the DMU data, through the analysis of the product structure, geometry and meta-data searching for a full product description [19]. The completeness of a DMU has to take into account the different phases of the DMU. It supposes a certain level of design maturity.

Assuming the DMU is built in the three phase geometrical approach described in Sect. 13.3.1.1 several methods can be used to calculate the completeness of the DMU. Below an example is proposed based on the number of interfaces each part of the product has. Furthermore any other convenient feature or meta-data of the DMU can be used:

1. Phase 1 completeness: as being in the phase 1, models in the product structure are expected to roughly claim the space where design and interfaces with other systems are expected to be positioned. Therefore, phase 1 completeness can be defined as the amount of planed interfaces per part as a part of the whole and the amount of existing models as a part of the expected.
2. Phase 2 completeness: in the phase 2 some parts may have shifted to assemblies depending on their complexity and functionality. In phase 2 it is important to define all interfaces geometrically and form and function of the part or assembly. The completeness of the DMU in this phase would be then the amount of well-defined interfaces per part or assembly as a part of the whole and all the parts of models that meet already phase 2 requirements as a part of all model expected.
3. Phase 3 completeness: in the third phase the design has to be in its final stage of definition. The configuration of the product has to be taken into account. The completeness aims to calculate if for the configured specific product, all parts or assemblies are in phase 3.

13.4.6 Automation of Processes

The advantages of creating a DMU is the availability of a virtual, digitalized environment that fully describes a real system that can then be simulated [20] and routine checks, formerly done manually can be either done within the digital environment or digitally planed for real missions done by humans or robots [21] with minimal user assistance.

The DMU is a collection of geometrical digital information plus its metadata than can be digitally manipulated using software algorithms to perform tasks that manually would be time consuming, would only be done in a few samples, and do not guarantee a reliable repeatability. For this reason DMU is called *Repeatable Digital Validation* too.

13.4.6.1 Boxing

Boxing is a geometrical filter of DMU parts that are contained in a spatial shape, mostly a three dimensional quadrilateral made of right angled sides called a box, or within a sphere. In the first case the box is defined with two opposite corner points, in the second with a point defining sphere center and its radius. The most common geometrical filter used is the method by defining a box thus giving the name to the method. Boxing is an efficient method to allow loading a specific reduced DMU saving loading time and at the same time giving the user only the information needed, with less straining of the computer deriving in higher performance.

CAD systems offer out-of-the-box solutions for boxing solutions but these usually involve native CAD data and are prone to crash or have extremely long calculation times and depending on the filtering parameters the filter may not work to the satisfaction of the user. Therefore, it is very useful to have an automated box creation which is based on an analysis of the product structure and the bounding box of the geometry attached to it.

13.4.6.2 Wet Surface Filtering

Wet surface filtering is another geometrical filtering method that aims to create a dedicated DMU with only parts of the product that can be seen from an external observer. This DMU is useful for sales purposes, extraction of data for 3D printing etc. Wet surface filtering methods vary from using the opposed algorithm to boxing: filtering out all parts inside the box, to deactivating unseen features in the parts directly from CAD system [22].

13.4.6.3 Specific DMU Generation

Cable harnesses positioning and design have an especial place in the priority of product design. Cable harnesses have very specific and standardized requirements usually coming from formal authorities, at the same time the flexible nature of cable harnesses obliges harnesses in most cases to adapt its routing to existing design. A DMU based harnessing is ideal to adapt the pros and contras of its design. The design of cables obeys very specific geometrical rules e.g.:

- Perpendicular progression of the cable into interfaces,
- Dripping of water from the cable into the interface or connectors,
- Catenary course of the cable due to gravity,
- Clearance of certain routing types to other routes or to systems transporting water or damp to avoid interference or eventual electrical accidents.

These constraints make an automated or half automated cable harnessing possible, using the routing constrains as parameters for a spline from which a tube representing the cable is derived [23].

13.5 Available Digital Mock-up Technologies

Almost each CAD/PLM vendor offers his own or partner solutions for DMU which are mostly embedded in its PDM or CAD system.

13.5.1 Vendors

For visualizing and using the advantages of a DMU big CAD/PLM vendors offer a broad spectrum of out-of-the-box solutions that companies can then customize for their own needs. These solutions are found as Visualization Systems or Digital Review Systems, all offering functionalities to view 2D and 3D data formats. Vendors use mostly their own proprietary formats which allow extra functionalities for reviewing going a step further than merely visualizing. Most of them support open source or standard formats like STEP, JT or VRML.

Three out-of-the-box solutions from the main PLM vendors are shortly presented below.

13.5.1.1 Dassault Systemes

Within the Workbenches of CATIA V5 and of latest V6 DMU-Navigator allow a DMU visualization through an internal conversion of proprietary formats to the own simplified CGR or 3DXML formats. Information of the design intent like features are lost. DMU Navigator can also be used as a stand-alone solution while only importing known formats.

Further workbench in the CATIA world is DMU Space Analysis that offers functionalities of collision calculation (called clash analysis), cutting and clipping, comparison of 3D geometry. Combined with the Object Manager Workbench the calculation of measurements, gravity center and center of inertia are possible.

13.5.1.2 Siemens Teamcenter Visualization

In 2007 Siemens AG took over UGS PLM software package, now called Siemens PLM. The most important products are the high end NX series of CAx software, the mid-range option of SolidEdge combined with a package of PDM solutions under the name of Teamcenter. The DMU tool is the Teamcenter module for visualization (PLM Vis) that manages along with other commercial proprietary and open 2D and 3D formats the own JT and PLM-XML files for lightweight product visualization. PLM Vis is a scalable package offering from a free plug-in viewer, building up options and functionalities in collaboration to a full professional version.

This product's market success is inseparably linked with a massive market penetration of the format JT (see Chap. 11). While DMU became the core application of JT, it gathered a broad utilization in discrete manufacturing with over 4,000,000 licenses of software in use. JT was recognized by capability for the interactive display of very large assemblies (i.e. whole cars) [24].

13.5.1.3 PTC DIVISION Mock-up

The product visualization and DMU suite of the company PTC now goes under the name of Creo Elements/View. Creo is currently the PTC package of all software family devoted to CAx and visualization technologies. Since 2008 PTC offers a complete new software definition that replaces the former ProductView and DIVISION mock-up for processing faceted model as well as validating procedures. At the same time the program serves as a viewer as well as a complex validation tool with the special characteristic of being able to handle huge assembly groups: visualization of structures and navigation within huge assembly groups (thousands of parts and a data volume of several gigabyte) [25].

13.5.2 Customized Solutions

Vendor tools meet most of the client requirements. However, some have very specific needs that cannot be addressed by customization of out of the shelf vendor technologies. Therefore, manufacturers are obliged to tailor their own solution to satisfy their needs.

13.5.2.1 NetDMU

In the case of Airbus Industries, given the increasing amount of data and complexity and therefore collisions and collision follow-ups of the aircraft's DMU plus the obsolescence of UNIX workstation systems, the company found the opportunity to take into own hands the configured collision and clearance detection of its DMU. For this a 3 year project with a fully owned solution based on open source technologies was developed that efficiently calculates all collision and clearance checks needed overnight.

NetDMU is a three layered solution:

1. Data Processing: data that form the DMU from PDM, TDM and visualization systems is extracted, searching for the latest possible version of each assembly for all possible configured aircrafts of the actual programs. Product structure data is then broken down into discrete, smaller packages to avoid unnecessary

and result-less collision and clearance calculations. All possible calculation combinations are planned allowing the system to recognize calculations that can be performed only once but used in several result pools (e.g. parts A and B will collide against each other in several parts of the aircraft under different assemblies, built in different aircrafts). Finally geometrical data is finally converted into open-source, readable files that will be used into next layer.
2. Collision and Clearance Calculation: based on the thoroughly prepared data a collision and clearance calculation is started. To cope with huge amount of calculations needed NetDMU administers a large pool of client computers that calculate overnight and thus take advantage of not used computer power.
3. Data Management and Visualization: all data collected from the calculations is then again processed to condense repeated or reusable results. These results are then compared with existing ones feeding a results history data base. All results are then presented in a web-based interface, independent of platform or operation system, where all results can be viewed on an open-source format through an open-source web plug-in in any Internet browser.

13.6 Functional Digital Mock-up

The extension of the DMU to functional aspects (FDMU), is an attempt to create more powerful tools for product development. The virtual products stored in DMU shall be enriched by the information extracted from simulation and measurement tools which describes the functionality with respect to environment in an early phase of the product creation processes [26].

13.6.1 Background

Based on DMU, FDMU shall comprise the results of all simulations needed for full presentation of the behavioral system description. Literally spoken, FDMU extracts the data from all virtual models of a product and gives them a physical meaning. It makes the product function experienceable and facilitates the *physicalisation of data* by setting the physical effects in context of a product [27]. As a prerequisite, one has to ensure a deep interaction between visualization and numerical simulation with respect to product life-cycle management. FDMU application requires three basic components: a description of the geometry, a description of the behavior and a comprehensive visualization of the results with fine adjustable filtering capability (Fig. 13.7).

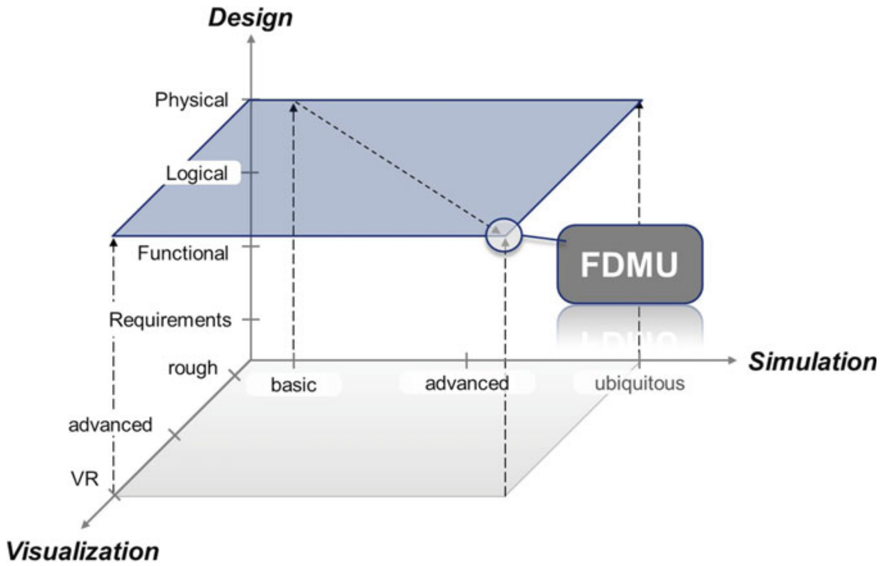


Fig. 13.7 Pillars of functional DMU

13.6.2 Existing Concepts

Nowadays, FDMU is rather a concept than a software product due to the complexity and challenge of its scope and diversity in the IT market. It is based on interoperability of design and simulation using a continuous information exchange between both types of models. The approaches can be basically divided into four groups based on different types of interoperability between geometry and simulation models [28].

A tremendous impact for all those FDMU approaches is given by Modelica Association with the recent functional mock-up interface (FMI) [29]. In this approach, corresponding commercial modeling and simulation tools are equipped with a common interoperability interface, listed in [30]. This interface exports the model output containing both the behavior and the input parameters of a simulation: those are practically the calculation method and the initial values of equations. A simulation tool with the FMI interface can then read these models to simulate a particular behavior. This approach is therefore particularly interesting in that several models and several simulations can be run in parallel (co-simulation), and, therefore, the overall behavior of a complex system can be represented and evaluated.

Nevertheless, all described approaches have significant weaknesses in representing the performance in the entire product context (e.g. a passenger car) with a short lead time (quasi online). While by zooming it is possible to online visualize each relevant detail within the DMU, the similar way is expected from experienced users for physical effects within FDMU. The FMI interface is already recognized in the automotive industry, its broad acceptance in the market is not assured yet.

13.6.3 Use Case

Typical use cases for FDMU always arise when several different physical effects appear on the tight space of a product. One such example is the passenger car, wherein the comfort of a living room is anticipated, which is affected by many vehicle-related as well as environmental influences. A central component of comfort optimization is the vehicle acoustics, which is considered in the complex NVH (noise, vibrations, harshness). While noise and vibration can be determined by appropriate experimental methods, harshness is a subjective property, and reflects human subjective impressions [27]. The psychoacoustic characteristics of a vehicle are a decisive factor for almost every buyer of premium cars.

No less important is to make the properties of the sound system in the vehicle in the way that an optimal subjective perception is achieved for every passenger at every interior variant. Here, both the position (driver seat, passenger seat, rear seat) and the fine adjustment of the seat position have great importance.

This is an area for FDMU that the acoustics of the vehicle interior is depicted for each vehicle variant for each passenger in each seat position to ensure that decisions on the design of the overall system can be represented entirely. The scenario describes a system for automatic volume control of (at least) two active sound sources. In this scenario, a person is in the vicinity of the two sound sources. His position, and, therefore, the distances from the sound sources can be changed. Both sound sources emit varying signals, e.g. music from a stereo system. In the area noise is present too, compromising the clarity of the music. Another part of the system is a microphone which detects the total sound pressure in the interior of the vehicle. Total sound pressure is composed of both the music and from the noise.

To support this analysis, an independent FDMU solution for use in the interior of passenger cars is developed based on CAD system CATIA V5 and the JT format paired with an appropriate viewer. The processing of the acoustic models is carried out with a specially developed solver module. Microsoft Excel supplies the Input/Output and low level standard calculations.

The user inputs data in the design table with the position and kinematics of seats. These data affect the behavior as well as the geometry in the selected use case. The user can change the CAD model if necessary, without influencing the behavior. The body height of the human as well as his seat position (driver, passenger, rear seat) are parameters, too. The list of parameters is closed by the geometrical position of the loudspeakers and the microphone in the context of a car interior.

One sheet in the result report contains the result for behavior with the required changes in the noise power of the speakers. A second sheet contains the description of the behavior in the use case. First, there is the distributed architecture of several frequency ranges with amplitude values measured by the microphone. These measurements represent the total sound pressure of all sounds in the application scenario. Secondly, the sound power levels of both speakers are registered across multiple frequency ranges. Along with the overall sound pressure measured by the microphone in the area and later determination of the position of the passenger, the

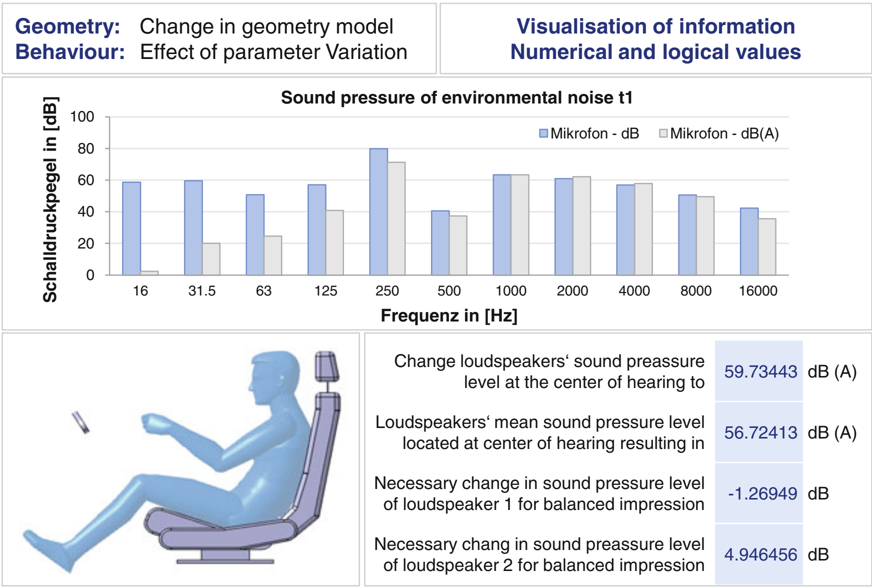


Fig. 13.8 Illustration of spaces with equal noise pressure level

sound pressure level of the noise can be determined (Fig. 13.8). It is desirable that the behavior of the system can also be shown directly on the geometry model, such as an application scenario for heat distribution on surfaces. It would illustrate the *goodness* of the sound field.

All means needed for visualization of such fields are available in the proposed concept, based on standard software. The implementation is necessary only by refining of the simulation algorithms. Similar procedure can be used in the demonstration of further interior equipment (e.g. sensors, heating, air conditioning, environmental effects, etc.).

13.6.4 Further Development

As an integrated standard FDMU system in the market offering of a software vendor seems unlikely at the present time, the further development must obviously run in two main directions.

First, the candidate applications must get better interoperability capabilities. For this purpose, the consistent implementation of FMI for each simulation tool would be very helpful und shall become one of the purchasing criteria.

Second, the loose coupling of individual components must be hierarchically controlled. FDMU is therefore distributed on four building blocks (Fig. 13.9): CAD/PDM system, hardware-in-the-loop (HIL)/software-in-the-loop (SIL) system,

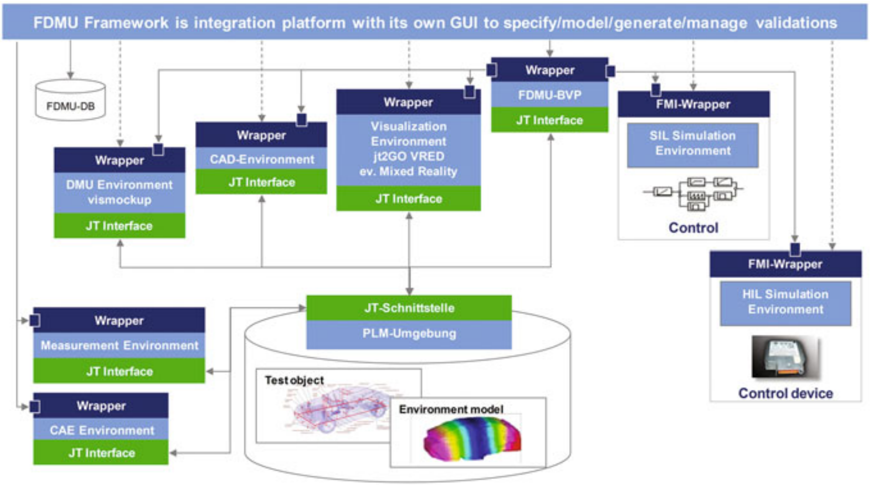


Fig. 13.9 FDMU architecture

DMU/VR system and, finally, the FDMU environment. To achieve a higher performance of the overall system, it is necessary to strengthen the component *FDMU environment* so that it assumes the user interface and the control of residual components. A similar application is described in [31]. This component shall also provide a template by which individual applications are inserted into FDMU system.

Finally, the open question remains of how the FDMU can be integrated into an enterprise-wide PLM concept to ensure the quick access, the data consistency and the broad data availability.

13.7 Industrial Case Studies

As DMU is mostly used in aerospace and automotive industry, this section covers three use cases which demonstrate the broad range of industrial practice.

13.7.1 Asynchronous Offshore DMU Validation in the Aerospace Industry

As described in Sect. 13.4.3.1 a collision detection aims to find design errors in the DMU so they can be corrected early enough in the process. The data collected from the collision detection can be also automatically analyzed but at a certain extent a human analysis is needed so the relevant collisions can be taken to an expert group to be discussed and reviewed.

The initial situation is a collision correction process done through automated detection. The collision detection delivers to each collision data regarding the colliding parts and their spatial position absolute and relative to each other allowing an automated filtering of known collisions. An on-site human evaluation of unknown collisions is performed by a follow-up team. Relevant collisions will be reviewed in an expert group. This procedure has the following problems:

- The main problem lies in the high costs of qualified personnel for the visually made human evaluation.
- Further problems in this situation are bound to the dynamical nature of the DMU being investigated. In this process there is no freeze of the geometry or product structure, this implies that in the time elapsed between a found collision, an analysis and the following review there can be changes in the DMU.

To tackle the costs of the follow-up a first optimization of the process is taken hiring an off-shore team of engineers that after intensive training of the follow-up team should realize the collision analysis and send the evaluated data back to the manufacturer.

In order to complete this strategy a package containing the frozen DMU (snaphot of product structure and geometry) is sent through secure data exchange to the off-shore team for the analysis. The package is stored into the suppliers file systems and the collision evaluation is started. The checks of relevancy are performed, in the case a collision is found relevant a screenshot of the DMU is made.

Through the same secure data-exchange channel the screenshots and the evaluation results are sent back to the manufacturer. The data of the off-shore supplier is then integrated into the collision databases of the manufacturer.

Collisions are then in design reviews solved.

This proposal implied itself new challenges and problems listed below:

- Data exchange of big data. The volume of the data sent to the supplier is significant challenging the bandwidth of both the manufacturer and the supplier. It has to be ensured that that big data packages were successfully exchanged. Problems with incomplete data-exchange caused extra delays in the evaluation thus augmenting the problem of data ageing and the completeness of the evaluated DMU.
- Data ageing. As described above the problem of data ageing is still a factor, the analyzed DMU can and may change during the evaluation time, making it difficult for the manufacturer to find the error in the DMU.
- Completeness. Trying to tame the problem of data size in order to reduce the complications in the data exchange, to find the balance of the amount of data needed for an evaluation became a challenge. In order to perform a good evaluation as much environment geometry as possible has to be taken into account. Too much information makes data packages even bigger and may deliver too much information that can become a security issue.
- Security. Although the data is send via secure channels to the supplier, the latter has sensible data of the manufacturer saved and stored in their file servers. Contractual

constrains define the usage and deleting of manufacturer data, nevertheless the manufacturer design information should be kept as secure as possible.
- Offshore. Personnel working in a different time zone reduces the overlaying work-time with the design department. Real time information flow is therefore constrained.
- Integration. An integration of the evaluation results has to be conducted in the manufacturers systems. To simplify this the supplier has to use the same tools and systems of the manufacturer recurring into, on one hand, a huge investment in licenses on the other hand installation of internal tools and data bases of the manufacturer into suppliers systems.

The problems above listed do not take into the account the quality of the analysis with a new team of inexpert co-workers. This approach was not a suitable alternative to the original situation. Costs and quality goals were not reached. New challenges emerged in terms of bandwidth, security and communication.

The second approach consisted of performing the clash evaluation on-site with the manufacturers systems, do packages of 3D PDF files with the results and exchanging them with an off-shore supplier.

Parts of the DMU involved in the found collisions, environment geometry and a 3D marker of the clash position are saved into 3D PDF files that are then sent via secure data exchange channels to the supplier. The data is checked by the supplier that sends the manufacturer an XML file with the inspected collisions and their relevance. The content of the XML file is mapped and transformed to be integrated into the manufacturers system.

This approach solves several of the problems from the original situation and those of the first approach (Fig. 13.10):

- Data Ageing: the data saved in the PDF file is not synchronized with the master DMU, therefore the problem of data ageing is partially solved. The dynamic changes of the DMU can always occur between the detection, analysis and reviewing but a frozen copy of the geometry exists to track back all problems.
- Completeness: the PDF file contains as much geometry as needed, the composition of the geometry can be decided by the manufacturer. The amount of geometry and, thus, the size of the PDF has to be well controlled in order not to create too big files that the supplier and/or manufacturer cannot open.
- Data Exchange: the solution with several relatively small PDF files solves the data exchange problem. Packages are sent and although data exchange problems may occur all packages that were successfully transmitted allow the supplier to start working. On the other hand, the packages sent back are very small XML files that pose no problem to the data exchange in terms of volume.
- Security: PDF files offer several ways of securing its content. For this solution a variant was implemented where the PDF file could only be opened by the supplier as long as they have a connection to a license server on the manufacturer's side. This gives the manufacturer the power to decide who, from where and when can open a PDF file and therefore define when the PDF file expires protecting its content.

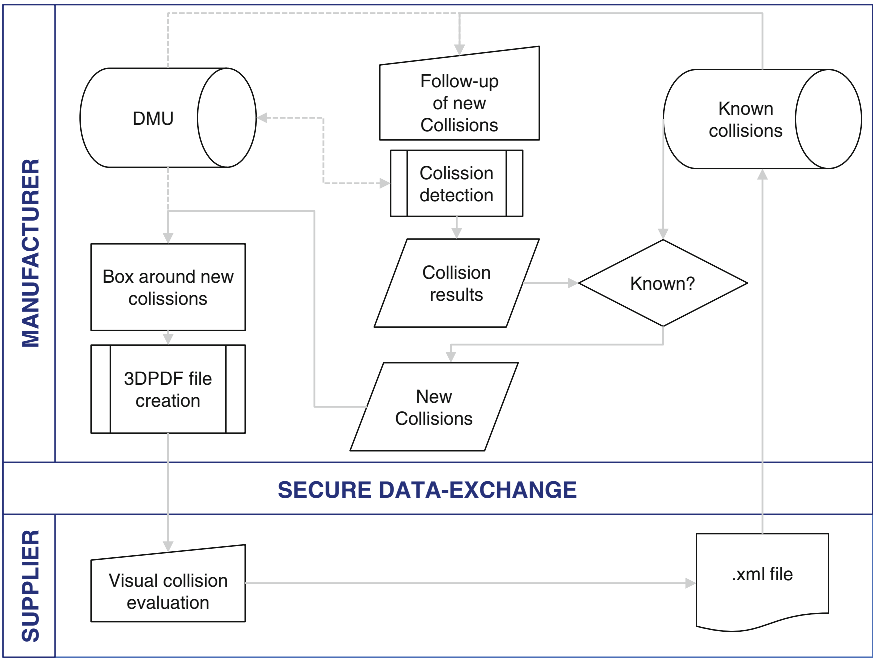


Fig. 13.10 Flowchart, asynchronous 3D PDF collision evaluation

- Integration: the integration of the XML file with the results pose no problem to the manufacturer.
- Off-shore time overlapping: this problem is not solved with this solution.

13.7.2 *Car Concept Validation in Automotive Industry*

Progress of DMU was driven by the automotive industry too. Complex products, short lead times, broad collaboration, high density of the package are all the reasons for the development and use of DMU. As shown in Sect. 10.6, Knowledge-based engineering (KBE) is an appropriate means for support of validation processes in the concept development of passenger cars, in particular in case of modular architecture and multi-brand product strategy. The concept development of a passenger car runs by creating a 3D master model and using many knowledge based assistants that support the designers in dedicated design tasks: silhouette of a car, design of wheel housing, pedestrian protection, gap tolerance area, ejection mitigation, barriers for bumper tests protection, seat belt numerical simulation [32].

In the area of design validation by DMU there is strong demand for automatic cross-section generation in 3D models. Creation and evaluation of conceptual designs takes place in 3-D CAD. Pre-defined 2-D design cross-sections of traditional package plans still form an indispensable part of the development process. Whenever new CAD software is introduced, the methods of interactive 2-D section derivation with special applications are preserved. Due to growing model sizes, design engineers have had to accept constantly increasing amounts of manual effort and waiting times for this purpose. The assurance of sustainable faultlessness —i.e. currency, accuracy, repeatability—for regular cross-section generation is becoming increasingly difficult with the growing number of projects.

To fulfill increasing process requirements, cross sections of all conceptual DMU data shall be generated fully automatically during the night. The automatic cross-section generator shall provide project coordinators with extensive configuration options for classifying the cross-section geometries for various graphic representations. Various benefits are expected: substantial time-saving for concept developers, obsolescence of manual cross-section maintenance and a significant increase in process quality.

To support this, a software tool called Automatic Cross-Section Generator has been developed, offering various configuration options for automatic processing of car concepts. It is implemented as an additional workbench with CAA V5 application programming interface for CATIA V5 and fully embedded in the CAD infrastructure. It consists of four main building blocks: configuration, pre-processing, sectioning and post-processing (Fig. 13.11).

Configuration is maintained via graphical user interface. It facilitates the setup of global options, section-job definition, continuous sections, special sections and section of reference cars (from previous or other current product families). Additionally it ensures typical administration tasks: classification/configuration of sectioning results, editing job definitions and defining sets of job definitions.

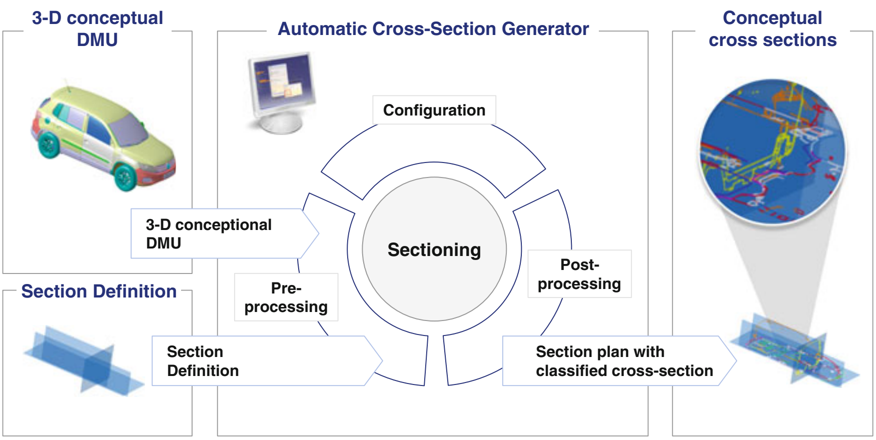


Fig. 13.11 Architecture of automatic cross-section generator [32]

The building block pre-processing prepares the operational sectioning carrying out the following tasks: analysis of 3D product and section definitions (valid product-structure, duplicate detection, check of metadata, valid input for sectioning process, space analysis) and modification recognition while minimized calculation time is achieved based on detection of modified geometry.

The main building block of the automatic cross-section generator is sectioning that processes the input CATIA package following the complex design rules. The input 3D conceptual DMU CATProduct remains unmodified, while cross-sections are created in the CATProduct containing the section plans. Report files are generated and put in pre-defined directory.

After the needed sections are generated, the resulting cross-section data is stored in the 3D CAD structure during post-processing. Modification of resulting cross section geometry is based on classification rules: structure, geometrical sets, properties, line type, line thickness and line color.

To leverage the operation and monitoring, strong optimization of loading and memory strategies is implemented. Batch operation is optimized by subdivision into different processes what ensures high performance as well as decrease of maximal memory usage. Realization of different report and protocol levels round up the user interaction.

Introduction of the automatic cross-section generator and batch operation on a high-performance server signifies a substantial time-saving for the concept developers. Thus, significant process quality improvement by automation and permeation of the whole vehicle project at predefined levels without user interaction are achieved. Resulting time and capacity savings are significant, while the move from manual to fully automated work has become possible.

13.7.3 Automotive Mixed Mock-up

With the new development platform Automotive mixed mock-up (AMMU), hybrid prototypes could be made available for the different engineering divisions at every development phase. Those prototypes combine the current stage of virtual development (DMU) as well as physical development (physical mock-up). This novel development platform provides a basis for new specification and validation approaches within the different development phases, ranging from the first digital drafts to the start of production in the plant [33]. Traditionally, digital and real processes ran consecutively. Initially, to ensure solutions were viable to put to real-world tests, early studies were conducted in a virtual environment during the development phase as well as mock-ups and part builds, such as seating configurations for ergonomic studies or complete vehicle prototypes, were then produced. Decisions were made either using models in the virtual world or directly affecting the real builds. MR linked these two worlds, the digital and the real, by embedding the virtual mock-up into real-life environment ensuring the assessment and visualization of modifications and new concepts as *mixed mock-up*.

MR first uses a camera to capture photos of a particular area of the prototype, such as the engine compartment (Fig. 13.12). The precise spatial relationship between the real-life and digital world is established using tracking systems, which precisely determine the current position of the video camera in relation to the overall product such as a vehicle. These video images are then displayed on a monitor and a variety of different components are *installed* virtually. Typical examples of application include positioning new electrical components such as wiring harnesses or control units, optimized heat shields or hydraulic functional units such as the brake force booster with corresponding lines. However, it is also possible to conduct installation tests with new engine configurations. In this case, the engineers assess whether a modified engine will fit in the existing installation space or if it can be easily mounted during the production process. Special spatial analyses are used by developers to assess the accessibility of individual components within the engine compartment to ensure that installation and maintenance work can be easily carried out on the vehicle at a future date. The first validation of virtual models is taking place in the real world with the use of MR.

This approach points out the capability of realigning respectively of adapting established processes (like maintenance and training activities) to new and uprising demands by means of allocating a clear and visual interface inbetween virtual and physical product development. Particular attention is paid to the information and knowledge transfer between the several organisational units, especially the experience and perception exchange in between virtual and physical development steps.

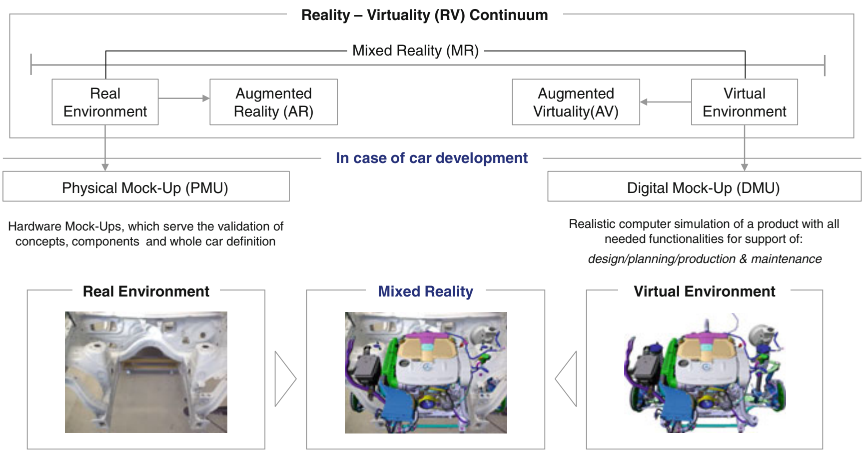


Fig. 13.12 Automotive mixed mock-up (AMMU) (reproduced with kind permission from Daimler AG © 2014 [33])

13.8 Conclusions and Future Perspectives

The DMU is a very useful tool for the product development and integration of stakeholders around a common goal to create competitive products. The future of the DMU is dependent on:

- The integration of information never meant for the mock-up.
- Development of networked virtual environments (NVE).
- Integration of DMU tools with other technologies.
- Usage of the DMU as a center piece of broader lifecycle strategy.

The perspectives of the DMU look forward to integrate information that originally was not meant for a mock-up and therefore was no requisite for the DMU development such as the aesthetic intentions of the design [34], concepts and inspirations, best practices, tips and tricks that may not be well documented in the PLM strategy.

The development of DMU usage goes hand in hand with the development and use of NVEs, how users are connected over different physical locations and the within build working spaces [34] and the capabilities of these environments to allow users interact with each other. The possibilities of interaction surpass the human face-to-face interaction. The advancements of technology pass over hurdle after hurdle of needs and goals of collaboration and communication [35], the challenge of the DMU is to integrate these technologies to improve the design and review process.

An interesting future perspective of the DMU is the impact of the Web 2.0, especially social media in the design process [36]. Product relevant information can be shared through social media similar functionalities that are or coupled with the PLM system or are part of it in the form of an extension. After this information has reached a certain maturity it can be then saved in the PDM system attached to the product structure [37] and hence enhancing the scope of metadata as part of the DMU. This information may be useful to the validation, review process or any other derived from the DMU giving a broader view of the context surrounding the geometry that is being visualized acting as a business tool that communicates past decision taking, context etc. Visualizing and incorporating such information is still an option rather not a challenge due to the vast growth of social media technologies and gaming in the past years crossing the problem of user friendliness of PDM systems across the enterprise [38].

The design information can be also done itself in a collaborative session closing the gap with the posterior design review while developing a technology of simultaneous real-time design where while reviewing designing is also possible [39, 40].

The DMU is an *as designed* description of the product and differing from the mock-up, does not always take into account changes that occur during the manufacturing or assembly process. Bend sheet metals, casting models and spontaneous last minute on site decisions at the assembly create an unavoidable difference between the DMU and *as manufactured* product. In those cases the DMU is not actual, mature or accurate. The flow of information back into the DMU is possible using 3D optical shape acquisition used for quality checks [41] comparing the DMU with the actual product at part by part level (see Chap. 12). The further step is to recognize the changes comparing the manufactured part with the design part in the DMU and in the case of a change generating new geometry with the changes while updating the geometry in the product structure and thus the DMU [42]. Although this step is today with constraints possible the practice is not widely used and the problem of closing the *as design—as manufactured* gap is not widely recognized.

References

1. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manage 7(2):116–131
2. Aoyama H, Kimishima Y (2009) Mixed reality system for evaluating designability and operability of information appliances. Int J Ineract Des Manuf 3:157–164
3. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manage 7(2):101–115
4. Mengoni M, Germani M (2006) Virtual reality systems and CE: how to evaluate the benefits. In: Ghodous P et al (eds) Leading the web in concurrent engineering. IOS Press, Amsterdam, pp 853–862
5. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manage 6(4):307–323
6. Valentini PP (2009) Interactive virtual assembling in augmented reality. Int J Interact Des Manuf 3:109–119
7. Jareño JJ (2009) Application of the engineering in the aeronautics. http://tv.uvigo.es/uploads/material/Video/5147/PRESEN-2009-12JJjare____o.pdf. Accessed 15 Nov 2013
8. Mart T, Cangelir C (2013) Lessons learned for better management of master geometry. In: Bernard A et al (eds) Product lifecycle management for society, IFIP advances in information and communication technology, vol 409. Springer, London, pp 712–721
9. Di Gironimo G, Patalano S, Tarallo A (2009) Innovative assembly process for modular train and feasibility analysis in virtual environment. Int J Interact Des Manuf 3:93–101
10. Voss T (2008) Untersuchungen zur Beurteilungs- und Entscheidungssicherheit in virtuellen Umgebungen. PhD thesis, Technische Universität München
11. Hudelmaier J (2003) Sichtanalyse im Pkw unter Berücksichtigung von Bewegung und individuellen Körpercharakteristika. PhD thesis, Technische Universität München
12. Dierßen S (2002) Systemkopplung zur komponentenorientierten Simulation digitaler Produkte. PhD thesis, Eidgenössische Technische Hochschule Zürich
13. Deuschl M (2006) Gestaltung eines Prüffelds für die Fahrwerksentwicklung unter Berücksichtigung der virtuellen Produktentwicklung. PhD thesis, Technische Universität München
14. Mendonça CH (2007) The system verification breakdown method. In: Loureiro G et al (eds) Complex systems concurrent engineering. Springer, London, pp 65–72
15. Verlinden J, Horváth I, Nam T-J (2009) Recording augmented reality experiences to capture design reviews. Int J Interact Des Manuf 3:189–200
16. Hrimech H, Merienne F (2010) Interaction and evaluation tools for collaborative virtual environment. Int J Interact Des Manuf 4:149–156
17. Ludwig L, Haurykiewicz J (2007) Collision checking analysis tool: discovering dynamic collisions in a modeling and simulation environment. Int J Interact Des Manuf 1:135–141

18. Mas F, Gómez A, Menéndez JL, José Ríos J (2013) Proposal for the conceptual design of aeronautical final assembly lines based on the Industrial Digital Mock-Up concept. In: Bernard A et al (eds) Product lifecycle management for society, IFIP advances in information and communication technology, vol 409. Springer, London, pp 10–19
19. Dolezal WR (2008) Success factors for digital mock-ups (DMU) in complex aerospace product development. PhD thesis, Technische Universität München
20. Zachmann G (2000) virtual reality in assembly simulation—collision detection, simulation algorithms and interaction techniques. PhD thesis, TU Darmstadt
21. Raffaeli R, Cesetti A, Angione G, Lattanzi L, Longhi S (2012) Virtual planning for autonomous inspection of electromechanical products. Int J Interact Des Manuf 6:215–231
22. Kanai S, Iyoda D, Endo Y, Sakamoto H, Kanatani N (2012) Appearance preserving simplification of 3D CAD model with large-scale assembly structures. Int J Interact Des Manuf 6:139–154
23. Valentini PP (2011) Interactive cable harnessing in augmented reality. Int J Interact Des Manuf 5:45–53
24. Handschuh S, Dotzauer R, Fröhlich A (2012) Standardized formats for visualization—application and development of JT. In: Stjepandić J et al (eds) 19th ISPE international conference on concurrent engineering, concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Springer, London, pp 741–752
25. Rakotomamonjya X (2007) Experimentation of an enterprise architecture in aerospace electrical engineering process. In: Loureiro G et al (eds) Complex systems concurrent engineering. Springer, London, pp 683–691
26. Balasubramarian B (2008) Entwicklungsprozess für Kraftfahrzeuge unter den Einflüssen der Globalisierung und Lokalisierung. In: Schindler V, Sievers I (eds) Forschung für das Auto von morgen. Springer, Berlin, pp 359–372
27. Fukuda S, Lulić Z, Stjepandić J (2013) FDMU—functional spatial experience beyond DMU? In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 431–440
28. Biahmou A, Fröhlich A, Stjepandić J (2013) Improving interoperability in mechatronic product development. In: Proceedings of PLM 10—international conference on product lifecycle management. Inderscience, Olney, pp 510–521
29. N N, Functional mockup interface (FMI)—version 1.0. https://www.fmi-standard.org/downloads. Accessed 15 Nov 2013
30. Blochwitz T et al (2012) Functional mockup interface 2.0: the standard for tool independent exchange of simulation models. In: 9th international modelica conference, Munich, 3–5 Sep 2012. https://trac.fmi-standard.org/export/700/branches/public/docs/Modelica2012/ecp12076173_BlochwitzOtter.pdf. Accessed 15 Nov 2013
31. Stork A et al (2009) FunctionalDMU: towards experiencing the behavior of mechatronic systems in DMU. Fraunhofer IGD, Darmstadt. http://www.igd.fraunhofer.de/sites/default/files/FDMU%20Pr%C3%A4sentation.pdf. Accessed 15 Nov 2013
32. Brüning HC, Liese H (2013) Reliable methods for the virtual car design process in the conceptual Development of passenger cars at volkswagen AG. In: ProSTEP iViP symposium. Hannover, 16–17 April 2013
33. Geißel O (2012) AMMU automotive mixed mock-up. Konzeption einer neuen Entwicklungsplattform für die Automobilindustrie. PhD thesis, Universität Stuttgart
34. Cheutet V, Léon J-C, Catalano CE, Giannini F, Monti M, Falcidieno B (2007). Preserving car stylists' design intent through an ontology. Int J Interact Des Manuf (2008) 2:9–16
35. Baladi M, Vitali H, Fadel G, Summers J, Duchowski A (2008) A taxonomy for the design and evaluation of networked virtual environments: its application to collaborative design. Int J Interact Des Manuf 2:17–32
36. Brown J (2012) Social business collaboration and the product lifecycle: combining the collaborative power of social media with PLM. Tech-Clarity Inc. http://www.tech-clarity.com/documents/Tech-Clarity_IssueinFocus_Social_Business_PLM.pdf. Accessed 9 July 2013

37. Doumit N, Huet G (2013) Fortin C (2013) The role of enterprise social media in the development of aerospace industry best practices. In: Bernard A et al (eds) Product lifecycle management for society, IFIP advances in information and communication technology, vol 409. Springer, London, pp 356–364
38. Huet G, Zeng Y, Fortin C (2012) Theoretical foundations supporting the implementation of complementary information structures across the life of a product. In: ASME 2012 11th biennial conference on engineering systems design and analysis. Nantes, France, 2 July 2012
39. Aronoffa M, Messinaa J (2007) Collaborative augmented reality for better standards. In: Loureiro G et al (eds) Complex systems concurrent engineering. Springer, London, pp 479–486
40. Dineva E, Bachmann A, Moerland E, Nagel B, Gollnick V (2014) New methodology to explore the role of visualisation in aircraft design tasks: an empirical study. Int J Agile Syst Manag 7(3–4):220–241
41. Raffaeli R, Mengoni M, Germani M, Mandorli F (2012). Off-line view planning for the inspection of mechanical parts. Int J Interact Des Manuf 1:1–12
42. Herlem G, Ducellier G, Adragna PA, Durupt A, Remy S (2013) A reverse engineering method for DMU maturity management: use of a functional reeb graph. In: Bernard A et al (eds) Product lifecycle management for society, IFIP advances in information and communication technology, vol 409. Springer, London, pp 422–431

Chapter 14
Modularity and Supporting Tools and Methods

Josip Stjepandić, Egon Ostrosi, Alain-Jérôme Fougères and Martin Kurth

Abstract The paradigm of modularity has emerged as a relevant way to meet customer requirements with a wide range of variety and customisation of products, from unique to standard ones. The modularity area is becoming increasingly multidisciplinary, which implies holistic and articulated concurrent engineering approaches. Modularity can intersect technical aspects with the business aspects. The use of modular technology has wide-reaching implications for any design and development company that undertake to use this paradigm. This chapter provides a framework for understanding the modularity in the context of concurrent engineering. It involves design for modularity as well as management of modularity. Theoretical and practical development of consistent modular methods, their implementation technologies and tools for mass customization and product configuration are examined. Some of the possible implications of these developments are presented from concurrent engineering point of view. The current trend is drawn toward usage and integration of different technologies such as advanced CAD systems, product configurators, agent-based systems and PDM systems. Three particular application areas with industrial use cases are presented. A discussion about research challenges and further developments closes this chapter.

Keywords Modularity · Modular design · Product variety · Mass customisation · Product platform · Product configurator

J. Stjepandić (✉)
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: josip.stjepandic@prostep.com

E. Ostrosi · A.-J. Fougères
Université de Technologie de Belfort-Montbeliard, Belfort 90010, France
e-mail: egon.ostrosi@utbm.fr

A.-J. Fougères
e-mail: alain-jeromefougeres@utbm.fr

M. Kurth
RAYCE EURL, Lörrach 79539, Germany
e-mail: martin.kurth@rayce-lab.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_14

14.1 Introduction

Many sectors of human existence are tightly connected with the term of modularity. Besides techniques, modularity is intensively used in education, science (mathematics, informatics, psychology, biology and linguistics), media science, management, organization, financial services and the public administration. Modular products accompany practically a person throughout his life. As a commonly known artifact, the entire Web has a modular structure, composed of independent sites and pages, and each webpage itself is composed of elements and code that can be independently modified and can interact via clearly defined interfaces [1].

Modularity as an engineering and management domain has become relevant in the 60 s of the twentieth century through the design of the first modular computers. Through the development of concepts and a body of knowledge modularity has become an area worthy of study in its own right. It can be considered that the roots of modularity can be derived from human cognitive abilities [2]. In the 1980s Fodor revived the idea of the modularity of mind, although without the notion of precise physical localizability. According to Fodor modular (cognitive) systems fulfill certain criteria:

1. Domain specificity: modules only operate on certain kinds of inputs
2. Informational encapsulation: modules need not refer to other psychological systems in order to operate
3. Obligatory firing: modules process in a mandatory manner
4. Fast speed: probably because modules are encapsulated and mandatory
5. Shallow outputs: the output of modules is very simple
6. Limited accessibility
7. Characteristic ontogeny: there is a regularity of development
8. Fixed neural architecture.

These criteria are also valid in equal measures for modular technical systems. The precise definition of product modularity is provided by articulating a product system modularity construct in the domain of tangible assembled artifacts [3]. It focuses product modularity to the criteria of component separability and component combinability. This definition is finally related to other definitional perspectives synthesized by a literature review: component commonality, function binding, interface standardization, and loose coupling. The nomological network of the product modularity construct is derived from it and subject to further validation.

In context of concurrent engineering, modularity combines technical aspects with business aspects, both from a qualitative and a quantitative viewpoint (Fig. 14.1).

Complex products can be understood as a network of components that share technical interfaces (or connections) in order to function as a whole. Component modularity is defined based on the lack of connectivity between components [4]. Technically (Fig. 14.1), it can be expressed with three measures: (a) how components share direct interfaces with adjacent components, (b) how design interfaces

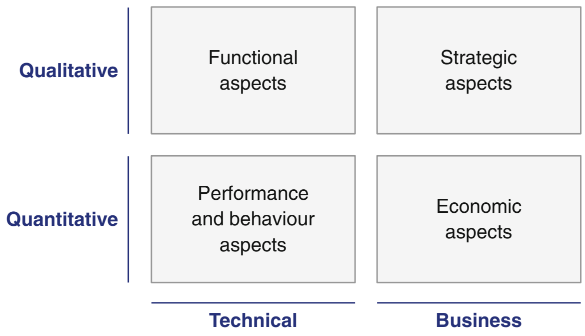


Fig. 14.1 Aspects and viewpoints of modularity

may propagate to nonadjacent components in the product, and (c) how components may act as bridges among other components through their interfaces. All three measures of component modularity have been identified for the product architecture of a large commercial aircraft engine. While trying to redesign components it was detected that the relationship between component modularity and component redesign depends on the type of interfaces connecting components.

Component commonality—the use of the same version of a component across multiple products—is being increasingly considered as a promising way to offer high external variety while retaining low internal variety in operations. As components influence to a greater or lesser extent nearly every process step along the supply chain, a multitude of diverging commonality problems is being investigated in literature. Representation of networks of components by graphs and the development of graph-based approaches have been applied as flexible and efficient means to a wide range of commonality problems [5].

Product-family design and platform-based product development also use the concept of modularity. Decision frameworks have been also introduced in this context to reveal a holistic view, encompassing both front-end and back-end issues such as product portfolio and product family positioning, platform-based product family design, manufacturing and production, and finally supply chain management [6].

From the business point of view (Fig. 14.1), modularization has three purposes: to make complexity manageable, to enable parallel work, and to accommodate future uncertainty [7]. The impact of modularity to the financial and organizational structure of an industry can be described with three aspects: (1) Modularity is a financial force that can change the structure of an industry; (2) The value and costs associated with constructing and exploiting a modular design are explored; (3) The ways in which modularity shapes organizations and the risks that it poses for particular enterprises are examined.

Modularization in enterprise leads, thus, to the disaggregation of the traditional form of hierarchical governance. The enterprise is decomposed into relatively small autonomous organizational units (modules) to reduce complexity and to integrate strongly interdependent tasks while the interdependencies between the modules are weak. The dissemination of modular organizational forms yields a strong process orientation: the complete service-provision process of the business is split up into partial processes, which can then be handled autonomously by cross-functional teams within subunits [8].

This chapter examines the main developments and implementation of modularity in the context of CE. In Sect. 14.2, the foundations of modularity are highlighted from design and management perspective. Approaches for support of modular design are explained and compared in Sect. 14.3. Subsequently, technologies and tools for modular design are introduced in Sect. 14.4. Three industrial use cases (automotive, aerospace, plant design) are described in Sect. 14.5. A discussion chapter gives insight into further development in field of modularity. Finally, an outlook is given with respect to the future of modularity from a CE perspective.

14.2 Modularity: Design and Management

In general, from a management perspective modularity is seen as a business strategy for efficient structuring of complex products, procedures and services with the objective to rationalize the enterprise [9]. A modular system (product, procedure, service) is, in contrast to monolithic systems, composed of separate modules, which satisfy Fodor's criteria and can be used in different product variants. It is possible to develop modules independently of each other and then bring them together in an integrated whole, a product, a procedure or a service on the market. Basically, a module (as a building block or black box) is interchangeable with another one. By fusing of different modules the range of products and services (solution space, range of articles) is enlarged. Opposite to an ad hoc reuse, this approach invests selectively in systematic reuse ability for later benefit.

14.2.1 Modular Design

Modular design combines both, the benefits of standardization and the benefits of customization. The advantages of modularity result from the reuse of parts and the repetition of operations. Hence, higher quality, lower cost of production, lower delivery time and easier spare parts procurement are expected [10]. Modularity facilitates collaboration and thereby helps to increase flexibility and to minimize economic risk. Since many modules already exist and can be purchased from an external vendor, the best module available (outsourcing) can be selected. Because of mutual independency a module is easier to design, produce, test, maintain and repair than a single more complex system [11]. As a result, a large product variance can be covered with a minimum quantity of modules.

The disadvantages of modular products are a limited selection within the supply range compared to a (entirely) customized product or service as well as high sensitivity to possible design errors in a single module or, even worse, in the interface between two modules. Furthermore, the function of each product variant can never be optimal, while some variants may be over-dimensioned. Consequently, product differentiation for the customer may be inhibited or lost [12].

Modular design considers functions, properties and interfaces of product constituents. Standard interfaces make parts interchangeable, thereby reducing the expenditure for the combination of different product constituents. Modular design usually involves the following processes: the identification of product architecture and reusable components (building blocks) from existing products, the agglomeration and adaptation of singular building blocks into modules to derive a new design, and assessment of product performance and cost. Modular product architecture is generated by deriving a rule base (scheme) for the mapping of product functions to physical components. For the utilization of modules comprehensive interfaces become crucial. A modular architecture is distinguished from an integral architecture by the way functional elements are mapped to physical components: it has a one-to-one mapping (design logic) from functional elements to physical components of the product. Three basic types of modular architecture are defined, namely slot, bus and sectional, according to the interfaces between components [13].

Platforms as a special expression of a modular design are of particular relevance for an industrial practice. A platform is a standardized base product with fundamental functions and properties of the total product, on which a variety of similar products can be efficiently built by using subsystems, modules and components. In the platform the architecture and the interfaces to optional elements are included, which are used for differentiation of the end products [14].

14.2.2 Mass Customization, Variety and Configuration

Under the term "Mass Customization" a business strategy is defined that utilizes modular design for complex offerings of products and services that are configured on demand to achieve the best fit with customer-specific needs [15]. Product variety and customization are provided through flexibility and quick responsiveness ("Configure-to-Order"). Mass customization joins two concepts that are usually supposed to be opposite: mass production and customization including two approaches: mass and craft (single-piece) production. Mass production manufactures low cost products by reaping the benefits of standardization and scale economies. On the other hand, craft production assumes a high level of individualization since the products are tailored to specific customer requirements.

Opposite to entirely individual products from special-purpose production customization options are allowed for allocated product structure areas only. Similarly, the processes of order processing are highly standardized, in order to quickly and cost-efficiently fulfill individual customer requirements. One of the reasons to establish mass customization is the replacement of the strategy build-to-stock (BTS) which refers to products that are built before a final purchaser has been identified, with production volume driven by historical demand information. BTS becomes inflexible with new market demands. To prevent costs expansion by large finished goods inventory, an introduction of a build-to-order (BTO) strategy is necessary to become a mass customization manufacturing enterprise [16]. By moving the customer order decoupling point (CODP) upstream in the value creating chain, the customer influence is increased and the response time shortened. In case of the automotive industry it realizes the delivery to the customer of a bespoke vehicle 5 days after placing the order [17].

Product structures of customized products must be thoroughly adjusted for specific customization options by adopting entirely individual components that are specifically created besides of standardized and configurable modules. Generally, a fixed and a variable area of product structure can be identified, in which mandatory and optional spaces are foreseen for individual implementation. In such a structure, technologies and tools as follow in Sect. 14.4 are useful, supported by CE technologies like Knowledge-based Engineering (see Chap. 10) and Digital Mock-up (see Chap. 13). Management of such a flexible, change-robust product structure is predetermined in a superordinate PLM concept (see Chap. 16).

Mass customization heavily affects all phases in the product creation process. Product customization is usually supported by configuration systems (see Sect. 14.4.2). Generic conceptual procedures for designing such systems are important for mass customization. These procedures involve analysis and redesign of the business processes, which can be supported by a configuration system, analysis and modeling of the company's product portfolio, selection of configuration software, programming of the software, and implementation and further development of the configuration system [18].

14.2.3 Modularity from a Management Perspective

By now, modularity has become a basic irreplaceable development methodology inside the product strategy for a variety of technical products planning based on market research and correspondent forecast. Individual products are not developed anymore, only whole product families or product spectra. A later change on this strategy would be very expensive and could implicate massive negative effects [19]. Development of new modules is essentially a new innovation process.

Modularity seems counter-productive, when selective distinctive features are the reason to buy a product. When customers focus on elements, like styling, haptics, or specific colours creative freedom is necessary. In such cases modular design is not applicable, because investments in modular design outweigh the efforts to create a user-specific product of which the number is often very small.

With the introduction of modular design, project control has to be adapted when focused on individual projects. The funding of the development of individual components has to be increased for every development project for a modular product family. The integration of different product variants does not come with any monetary benefits if it is not organized through a holistic controlling approach [20].

This approach enables the assessment of modular product families as well as their holistic management based on the new modularity-balanced-score-card (M-BSC). Additionally, the different perspectives from production, development, marketing and sales need to be integrated.

Cost schemes of modular products can be established by decomposing the product family into generic modules to support cost calculation [21]. The candidate modules that are closest to the customer requirements can be retrieved by computing a similar degree of the case module and the target module for the same module model quantitatively. The search space is restricted for generic modules. The cost structure of the different types of modules is analyzed, i.e., the basic module and customized module, and the different cost estimation approaches are applied to different types of modules. The product cost can be accumulated with cost of modules in each level of the modular product family progressively.

14.3 Methodical Support for Modular Design

Modularity is achieved by partitioning information into visible design rules and hidden design parameters. It is beneficial only if the partition is precise, unambiguous, and complete [9].

The visible design rules ("visible information") are basic decisions that affect subsequent design decisions. Ideally, the visible design rules are established early in a design process and communicated broadly to those involved. Visible design rules consist of three categories:

1. An architecture, which specifies system modules and their functions.
2. Interfaces with description of module interaction (fit, connect, communicate).
3. Standards for testing a module" conformity to the design rules and for comparing performance of competing modules.

The hidden design parameters ("hidden information") are decisions that affect the design only within the local module. Hidden parameters can be used late and changed whenever necessary.

Common attributes of modular products are defined as follows [16]:

1. *Commonality of modules*: Components or modules are used at various positions within a product family.
2. *Combinability of modules*: Products can be configured by combining components or modules.
3. *Function binding*: There is a fixed allocation of functions to modules.
4. *Interface standardization*: The interfaces between the modules are standardized.
5. *Loose coupling of components*: The interactions between the components within a module are significantly higher than the interactions between components of various modules.

In modularization the combinability of elements is increased when starting from an existing production structure [22]. Thereby, through reduction of interdependence a larger amount of product variants can be generated from a given number of element variants. Another opportunity concerns reduction of the level of integration. This can be achieved by interface standardization.

There are various methods to support modular design like axiomatic design (AD), functional modeling, design structure matrix (DSM), modular function deployment (MFD) and variant mode and effects analysis (VMEA), which can be also used in combination with an architecture development process [23]. The interdependence and the level of integration of components are a measure of modularity. Classification of various product modularization methods and the comparison of methods in several application areas (product variety, product generation and product lifecycle) have shown that the generation of modules depends on both the chosen method and the weighting of different criteria [24].

14.3.1 Axiomatic Design

Axiomatic design (AD) is a systems design methodology using matrix methods to systematically analyze the transformation of customer needs into functional requirements, design parameters, and process variables [25]. Hereby, the attempt is made to build on the development of new products based on a system of axioms, which are based on mathematics or physics sciences. The formalization is supposed to lead to the design of technical systems. Starting from two axioms (the independence axiom and the information axiom) a system of theorems is set up.

AD contains four domains: customer domain, functional domain, physical domain and process domain. Each domain has its corresponding design elements, namely, customer attributes (CAs), functional requirements (FRs), design parameters (DPs), and process variables (PVs). The zigzag mapping among domains follows the process of product design. The independence axiom, which demands maximizing the independence of the functional requirements, is used to judge the rationality of design. The information axiom, which demands minimizing the information contents of the design, is used to select the optimum design.

The mapping process can be expressed mathematically in terms of the characteristic vectors for functional requirements (FRs) and design parameters (DPs) that define the design goals and design solutions. The relationship between the two vectors can be expressed by the design equation for product design as following:

$$\{\text{FRs}\} = [\text{A}]\{\text{DPs}\}, \tag{1}$$

where [A] is a matrix defined as the design matrix that characterizes the product design. Ideally, it's a square matrix (when the number of FRs is equal to the number of DPs). When the design matrix is either diagonal or triangular, the corresponding design satisfies the independence axiom ("uncoupled design" resp. "decoupled design"). Any other form of design matrix is called full matrix and will result in a coupled design that is not beneficial. The uncoupled design indicates that the design tasks are mutually independent, and can run concurrently shortening the development time significantly. The decoupled design indicates that the design tasks should be processed by sequence so that the whole process can be managed effectively.

The modular design based on AD runs in three major steps:

1. *Analysis of product using AD*: The fundamental functions requirements and design parameters are decomposed into levels of sub-function requirements and sub-design parameters resulting the functional-structure model and a full design matrix.
2. *Module definition*: Product design is implemented agglomerating the modules from the singular functions and the components in the functional-structure model.
3. *Reconfiguring the sequence of modules*: The relationships between modules are uncoupled or decoupled on the basis of AD. The uncoupled relationships mean that these modules can be carried out simultaneously and the decoupled relationships mean that these modules must be performed in sequence so that the effect of former modules can be considered and the iterations of design can be reduced. In this step, we can get the module-junction structure diagram that indicates the design sequence.

Mapping of the modular product architecture follows the axiomatic design to maintain the cross-domain module independence of functional requirements. Aerospace and defense systems have been analyzed as complex systems with long lifecycles with particular attention to time and flexibility [26]. By dividing the design perspective into two domains, one long-term architectural and one short-term modular, and identifying an ideal product architecture for that situation, the design in terms of integral architecture and modular flexibility was optimized and the resolution of the issue with conflicting short and long term goals supported.

14.3.2 Functional Modeling

Functional modeling place product functions as a central idea for structuring of product into modules. The role of functional modeling is not only to clarify understanding of a design problem, but also to serve as core to the modular solution. Questions arise from these roles of functional modeling. *How should functions can be expressed or represented so that their structure can be used for modular design? What are rules or heuristics to identify modules from patterns in function structure?*

Functional modeling is based on the assumption that product functions can be decomposed into smaller, easily solvable sub-functions. The modeling of functions as operations on flows allows designers to represent product functions and to decompose these functions into chains of connected elementary sub-functions. The sub-functions in these chains are related by the flow of energy, material or signal passing through the product to form a functional model, known as a function structure. The product functions and sub-functions are described in a verb-object form. They are represented by a black-boxed operation on flows of energies, materials, and signals.

Modularity and the modular form of the product depend on the form of chains of connected elementary sub-functions. The modular design based on the function decomposition follows these steps [28]:

1. *Define product functions*. Product functions are defined as operations on flows of energies, materials, and signals. They originate from customer needs.
2. *Decompose product functions into sub-functions*. For each product function, decompose the flows of energies, materials, and signals. For each input flow, define a chain of sub-functions that transforms the flow step-by-step into an output flow. Because sub-functions are modeled also as operations on flows, they can be standardized. A library of flows and sub-functions has been proposed by Hirtz et al. [27]. These libraries are called functional bases. A designer can decompose product functions into sub-functions chosen from the library of sub-functions (Table 14.1). Modeling of sub-functions as operations on flows implies a temporal order relationship between sub-functions.
3. *Integrate the chains of sub-functions*. The temporally ordered chains of sub-functions are integrated by connecting the chains. Thus, series of sub-functions, sequentially and/or in parallel, transforms input flows step-by-step into output flows. The decomposed product functions form now a function structure.
4. *Identify modules*. Groups of sub-functions related by flows can be observed in the function structure of the product to form modules. This observation has led to the formulation of heuristics to identify modules. Stone and Wood [28] proposed three heuristics based on the three patterns that a flow can experience: 1) a flow may pass through a product unchanged, 2) a flow may branch, forming independent function chains, or 3) a flow may be converted to another type. Figure 14.2 shows the overall function of an electric wok with its function structure with heuristically identified modules [28].

This method provides a systematic technique for identifying modular patterns at the functional model stage. This method also allows product architecture decisions to begin at a much earlier stage, i.e. at the functional model stage. Once modular patterns in functional structure are defined, the design of alternative layouts and components become more straight forward tasks. Optimization techniques may then be applied to find optimal modular product.

14.3.3 Design Structure Matrix

The design-structure-matrix (DSM) is a simple and meanwhile commonly used tool for the visualization of complex connections in systems in a compact and easily adjustable format. This facilitates their analysis. The systems can be both technical and social (e.g., organizations) [29].

Table 14.1 Library of sub-functions [27]

Primary	Secondary	Tertiary
Branch	Separate	Divide
		Extract
		Remove
	Distribute	
Channel	Import	
	Export	
	Transfer	Transport
		Transmit
	Guide	Translate
		Rotate
		Allow DOF
Connect	Couple	Join
		Link
	Mix	
Control	Actuate	
	Regulate	Increase
		Decrease
Magnitude	Change	Increment
		Decrement
		Shape
		Condition
	Stop	Prevent
		Inhibit
Convert	Convert	
Provision	Store	Contain
		Collect
	Supply	
Signal	Sense	Detect
		Measure
	Indicate	Track
		Display
	Process	
Support	Stabilize	
	Secure	
	Position	

The DSM is represented as a square N × N matrix, mapping the interactions among the set of N system elements. Depending on the type of system being modeled, DSM can represent various types of architectures (product, process, organization, multiple domains). For example, to model a product” architecture, the DSM elements would be the components of the product, and the interactions would

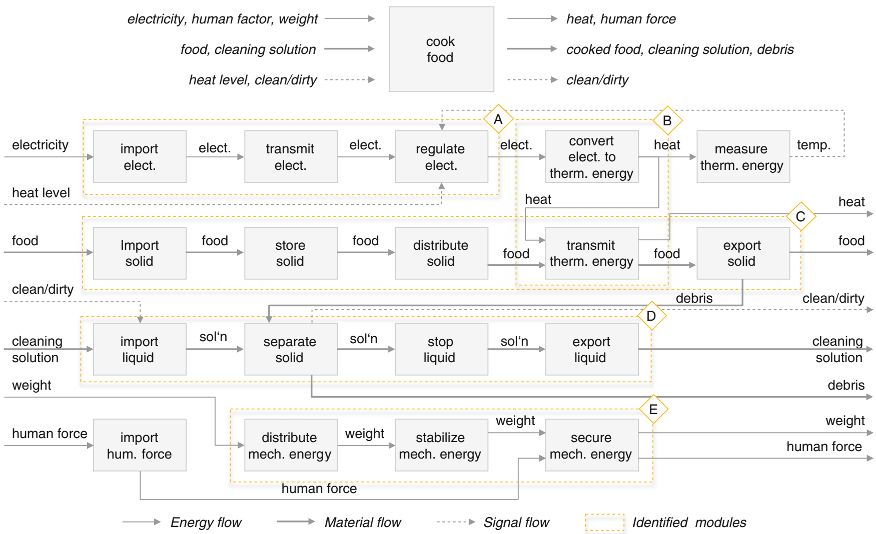


Fig. 14.2 An overall function of an electric wok with its function structure with heuristically identified modules [28]

be the interfaces between the components. Using the advanced three-dimensional DSM (DSM^{3D}), designers are able to highlight the differences among singular members within families of products, modules, and interfaces.

The DSM can be illustrated by the modular design of a single-use camera [29, p. 75] (Fig. 14.3). Through marks outside the diagonal (21, 21–15, 15) it is shown, that one component in the system is dependent on another one. Reading across a row reveals what to what other elements the element in that row provides outputs, and scanning a column reveals from what other elements the element in that column receives inputs. In the example, element 11 (inner lens plate) provides input for the elements 9, 10, and 13, simultaneously. Due to the symmetry, inter-dependent elements can be recognized: 9 needs 11 at an early stage and at the same time 11 is providing output to 9.

In this use case five Kodak single-use cameras were dissected. By comparing component interactions across the five camera models, interfaces were categorized as common (occurring in all five), variant (occurring in some of the five), or unique (occurring in one of the five) and colored (red, blue, yellow, respectively) into the matrix in Fig. 14.3. Based on interaction among components and appropriate clustering, five modules were identified indicated by square borders in the DSM. The last component in the list (structure) is related to many other components, as indicated with the many colored squares along the bottom of the DSM. This bus-type component is strategic because there is an opportunity to use common interfaces to save costs and better handle diversity.

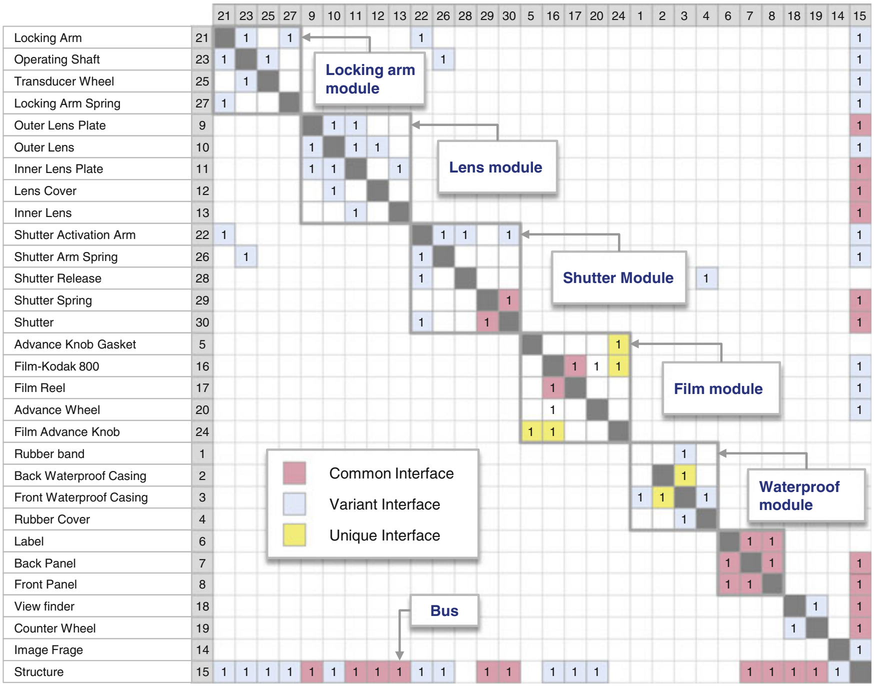


Fig. 14.3 Clustered DSM for modular design of a camera [18]

The DSM enables designers to study basic aspects of product architecture such as bus, mini-bus, and strength of physical interactions. It also helps to investigate the architectural distribution of modules and interfaces.

The DSM shows that many interfaces are variants. Interfaces and modules having instantiation in multiple products are named respectively cross-interfaces and cross-modules. By identifying cross-interfaces and cross-modules, which are beneficial from a cost saving point of view, designers should avoid variant interfaces (colored blue) that provide diversity and incur additional cost. In the case of variant cross-modules, the DSM indicates that it is necessary to develop a common cross module with common interfaces.

A small number of computer software applications incorporate dependency structure matrices in particular for software development. The DSM knowledge is maintained by the DSM Community, an interdisciplinary expert group [30].

A four-step product module identification approach by combining AD and DSM is proposed in [31] (Fig. 14.4). The overall pertinence DSM of DPs is obtained in first step, which assembles the three aspects of pertinence relationship of functionality, structure, and manufacturing process. The similarity DSM of DPs from manufacturing process is built in the second step. Based on the two steps above, the overall pertinence DSM with similarity DSM is aggregated and the overall

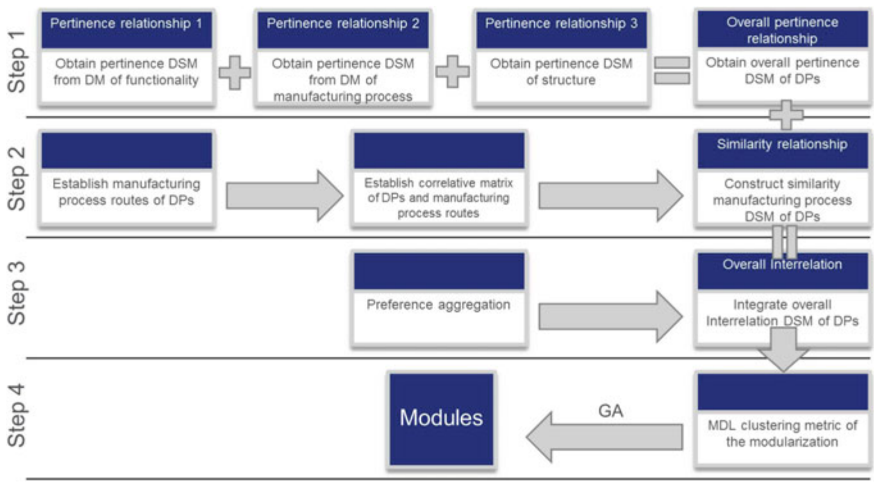


Fig. 14.4 Product module identification based on AD and DSM [31]

interrelation DSM of DPs is obtained in the third step. The overall interrelation DSM is clustered and modules generated by using genetic algorithms and minimal description length in the fourth step.

14.3.4 Modular Function Deployment (MFD)

As a further occurrence of a matrix method, the modular function deployment (MFD) approach was developed by Erixon [32] with the target to facilitate the development of a "robust" production program that is easily adaptable to the varying requirements. It is based on the quality function deployment (QFD) methodology, which is expanded with the modularity concept. The MFD approach introduces dedicated criteria ("module drivers"), which compile a business strategy into a framework for modular product design. Module drivers yield the basis for a systematic evaluation of technical solutions for a given product, based on an accommodated product structure.

MFD consists of five major steps (Fig. 14.5):

1. *Clarify customer requirements*: This step ensures that the precise design specification as functional requirements is derived from customer requirements e.g. by using QFD.
2. *Evaluation of functions and selection of technical solutions*: This step is also called "functional decomposition" because functions and sub-functions are identified and assigned to the technical solutions ("function drivers") to fulfill the functional requirements. For each requirement there are many possible technical solutions.

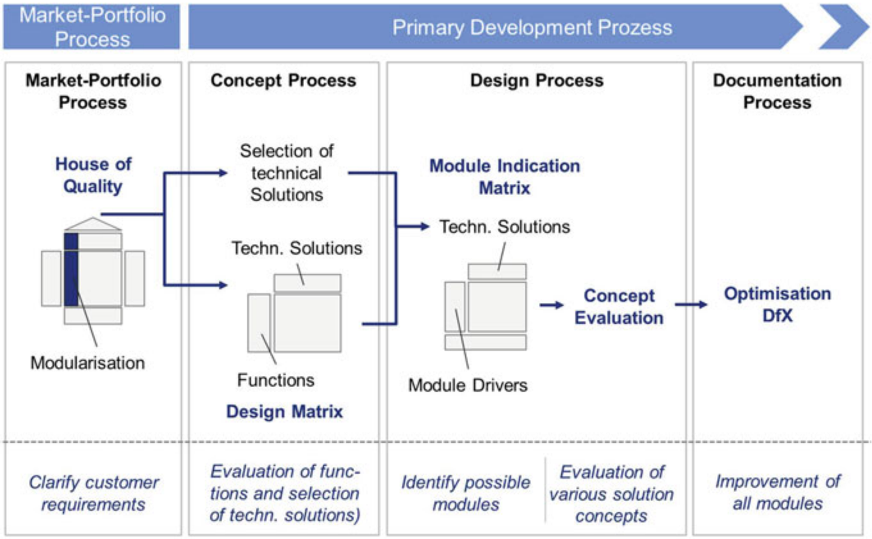


Fig. 14.5 Modular function deployment

3. *Identify possible modules*: The most important step contains the systematic generation and selection of modular concepts, in which the module indication matrix (MIM) is used to identify possible modules by examining the interrelationships between dedicated criteria ("module drivers") and technical solutions. A questionnaire is provided to support this activity.
4. *Evaluation of various solution concepts*: MIM also provides a procedure for investigating opportunities of integrating multiple functions into single modules by using the interface matrix. Herein, modules are first sorted in the expected assembly sequence. In this matrix is recorded which module are coupled by which kind of interface (so called "hamburger" or "base part assembly"). The expected effects of the redesign can be estimated and an evaluation can be provided for each modular concept.
5. *Improvement of modules*: The singular modules are improved—independent on each other by using appropriate ranking methods (e.g. Rank Order Clustering).

The overall understanding of MFD is that improved product modularity facilitates the entire flow of information and materials—from development and purchasing to logistics and delivery. The focus lies on the combination of function assignment and economic criteria ("module drivers"). It is a modularity shaping method, which integrates different weighed goals and produces a modular product architecture. Thus, multidimensionality is its special characteristic.

14.3.5 Variant Mode and Effects Analysis (VMEA)

The Variant Mode and Effects Analysis approach (VMEA) helps to depict the impacts of product variants in all units of the enterprise from definition of the product program to distribution. It includes an evaluation of target costs and discovers cost-saving potential by eliminating product variety that is not customer-perceived.

The VMEA is divided into three different levels; (1) basic VMEA, in the early design phase, when we only have vague knowledge about variation; the goal is to compare different design concepts, (2) advanced VMEA, further in the design process, when we can better judge the sources of variation, and (3) probabilistic VMEA, in the later design stages, where we have more detailed information about the structure and the sources of variation; the goal is assessment of reliability [33].

Advanced VMEA is used to optimize modularity [34]. A systematic development of variety diversity is supported according to technical and economic criteria. Optimal product structures are determined by variation of parts and modules. The VMEA integrates business units like marketing, product program planning, product development, production and distribution and is executed through the following steps (Fig. 14.6):

1. Market-oriented evaluation and design of product functions and determination of target costs.
2. Derivation of design alternatives for the realization of a homogenous product structure.
3. Evaluation of design alternatives for technical and economic aspects and selection of variant solution.
4. Definition of the product program and its transfer to sales.

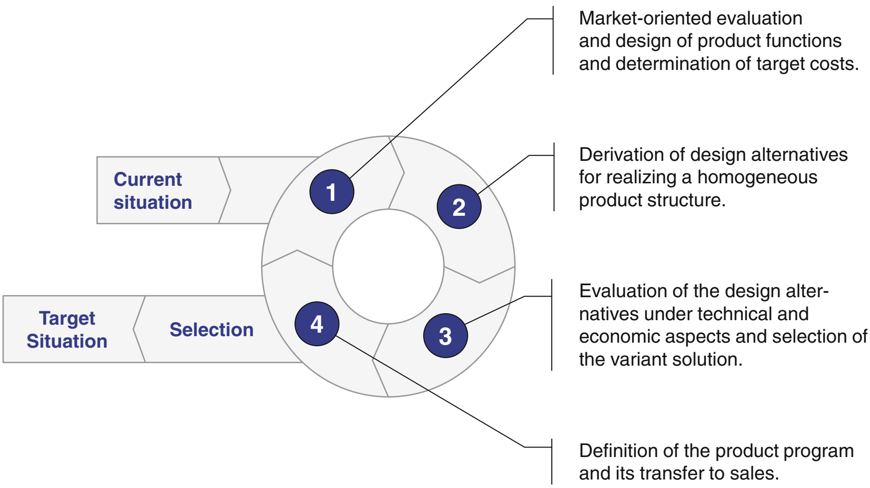


Fig. 14.6 Variant Mode and Effects Analysis (VMEA)

The primary benefit of VMEA is that the systematic preparation of multiplicity information and the graphic presentation of the development of variants can support the required communication between the units involved in VMEA: marketing/sales, development/design, work preparation/production and budgeting/controlling. Thus, the diversity of variants is known for future products and unneeded reserves concerning performance and flexibility of the equipment can be avoided.

14.4 Technologies and Tools for Modular Design

Currently, manifold technologies and tools are offered to foster modular design:

- Select pre-defined components and assemblies (standard parts library/catalog),
- Configure the product based on customer demand (product configurator),
- Facilitate the saving and selecting of knowledge of modular products (agent-based approach) or
- Facilitate definition and maintenance of general product structures, so called "150 % product structure" (PDM system).

These technologies have been developed independently and support specific aspects of modular design. They can provide optimal functionality by mutual integration and interaction with other systems (e.g. ERP). If their mutual customization is conducted thoroughly, all characteristics of modular design can be mapped.

14.4.1 Standard Parts Catalog/Library

The module standard parts catalog, which exists in every modern CAD system, is an easy way to foster product modularity, not only because it provides standard parts (according to national and international standards), but also because it facilitates the insertion of external part libraries and carry-over parts as a CAD functionality. In this case, the designer's activity is limited to the selection, insertion and validation of such parts in an assembly.

Based on this basic CAD functionality, enterprise part library solutions have been developed, which facilitate the allocation, administration and integration of external part libraries (e.g., which are provided by component suppliers) as an alternative for individual (printed or digital) catalogues. Thereby, extent and selection of purchased parts can be limited and prioritized according to individual criteria by a central standardization department in cooperation with the purchase department (e.g., parts of component suppliers with the best quality certificate are preferred). This will lead to massive cost savings. As a result of such search a designer gets 3D and/or 2D models in the desired native or neutral format (STEP, IGES, DXF) plus metadata for down-stream applications.

The administrator has the possibility to maintain the database in real-time by considering the business activities of a supplier (e.g., extension of its product portfolio). Also, regional differences in delivery can be considered. In case a standard part is required, which is not available, a pre-defined work flow exists to automate the generation of requests and to provide tracing, justifying and monitoring the processing of requests, and for reporting nonconformities. In addition to a part library a search engine for 3D data can be used that locates similar parts in large, heterogeneous data resources within fractions of a second. It analyzes the 3D geometry of the parts and automatically extracts characteristic features and facilitates part reuse in such way.

14.4.2 Product Configurator

A product configurator is a multi-functional, commercial IT tool which serves as interface between sales and delivery in an enterprise. It supports the product configuration process so that all design and configuration rules, as expressed in a product configuration model, are guaranteed to be satisfied. The configurable products are usually characterized by the following properties [35]:

- each delivered individual product is adapted according to individual requirements
- products are pre-designed in order to accomplish a given range of individual requirements
- each individual product is specified as a combination of pre-designed components or modules
- the products have a pre-designed general structure
- no creative or innovative design is needed as a part of the sales-delivery process
- products are adapted by a routine, systematic product configuration process.

A product configurator is based on configuration software, which is able to map complex configuration rules with or without usage of a geometry kernel to create a CAD model in modular design. A product configurator implements formalized product logic, which contains all "If-Then" configuration rules and constraints. The usability of configurators has been improved also through the use of a compilation step in which the full space of valid configurations can be constructed. Product configuration tasks are shown in Fig. 14.7. The customer inputs his detailed requirements controlled by the user interface. The configuration problem consists in finding a feasible product that satisfies both the customer requirements and feasibility constraints. A configurable product is defined by a set of attributes or components whose possible values belong to a finite set of values and a set of constraints on these attributes, which specify compatible combinations of the values. A product, which meets the customer's requirements in the best way, is then selected. After validity check and cost analysis, the bill of material (BOM), CAD models, and finally, the bid are generated.

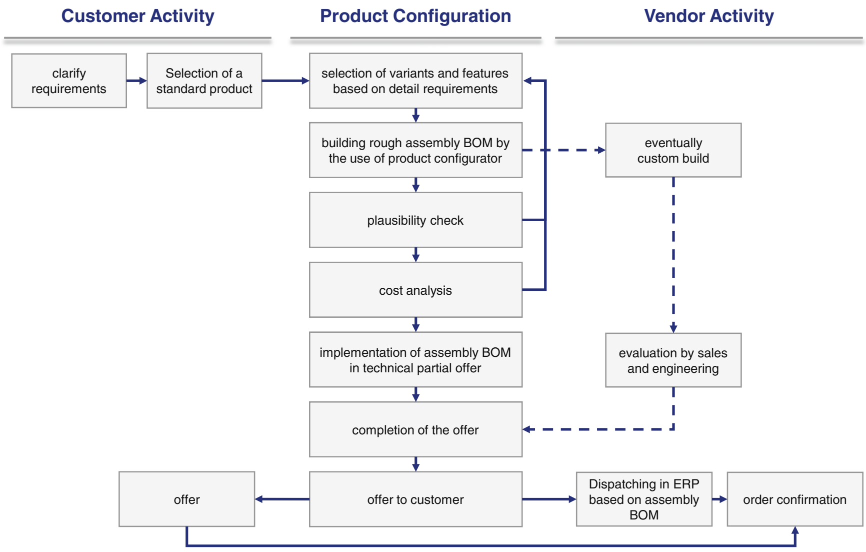


Fig. 14.7 Product configuration system

Today available commercial configurators have been designed as either enterprise resource planning (ERP)-centric or CAD-centric, depending on the need for a final CAD design activity (e.g. mounting specific adjustments) after the configuration is closed. By force of circumstance, as its function affects multiple core areas of an enterprise, a product configurator has to be integrated deeply with the involved IT systems such as ERP, product lifecycle management (PLM), CAX technologies. However the complexity associated with managing and synchronizing configuration master data across different applications such as ERP, PLM and CAX is an important barrier to the deployment of integrated product configuration. One study gives a comprehensive overview of the product configurators that are available on the market [36]. This plethora of commercial solutions indicates the maturity of this approach to support modular design as well as its deep market penetration.

14.4.3 Agent-Based Approach

Design for configurations is not only a structural problem but also a collaborative design problem [37]. Product configuration must explicitly consider different actors and their perspectives influencing the design of configurable products simultaneously. Uncertainty is also an integral part of configurable product modeling. Indeed, configurable product modeling must be able to deal with various unstable and imprecise requirements coming from customers, on the one hand, and some forms of uncertainty such as imprecision, randomness, fuzziness, ambiguity, and incompleteness, on the other [38]. Imprecision is caused by non precise or exact nature of design information. Randomness is referred to the lack of predictability or irrelevant or meaningless data considered as noise. Fuzziness is caused by incapacity to define the semantic of a variable rigorously, or the definition could be meaningless. Ambiguity is caused by indefiniteness in several interpretation of one word in the same language. Incompleteness is caused by the lack of information. Design process is, thus, the source of the uncertainty.

The *agent paradigm* can be applied to handle complex *uncertain problems* where global knowledge is inherently distributed and shared by a number of agents, with the aim to achieve a consensual solution in a collaborative way. Agents are autonomous and distributed entities capable of executing tasks either by themselves or by collaborating with other agents. An agent is a computer entity, located in an environment that it can perceive, in which it can act, possibly composed of other agents with which it can interact in an independent way. Fuzzy agents are proposed to solve distributed fuzzy problems [39] as well as to model the processing of the fuzziness of information, fuzziness of knowledge, and fuzziness of interactions in collaborative and distributed design for configurations [37, 40]. Structural problems of configuration are also formalized with the help of configuration grammars [41] and implemented in a grammar-based multi-agent platform [42]. An agent-based system called fuzzy agents for product integrated configuration (FAPIC) is developed for product configuration (Fig. 14.8) [43]. In FAPIC, each requirement,

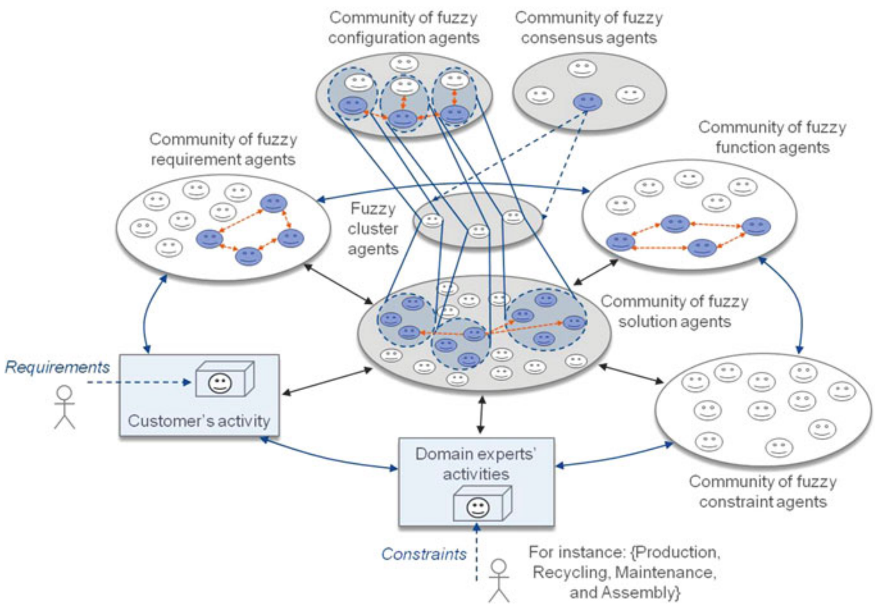


Fig. 14.8 Agent-based architecture of the FAPIC platform [44]

function, solution and process constraint is a fuzzy agent, with a degree of membership in each community of agents: requirement community, function community, solution community and constraint community.

Cooperative interactions between fuzzy agents are of two types: (1) Intra-communities interactions such as the interactions of fuzzy agents in the intra-community of functions and intra- community of solutions; (2) Inter-communities interaction between fuzzy agents of different communities such as the interactions: (a) between the requirement agents and the function agents, (b) between the function agents and the solution agents, (c) between the process constraints agents and the solution agents, and (d) between the configurations agents and the solution agents. Hence, there are discrete and continuous interactions between the agents of different communities.

In the first phase, FAPIC builds different societies of fuzzy agents, necessary for the configuration of a product. The agent-based system has been built according to five steps: (1) fuzzy agent-based communities building, (2) building interactions between fuzzy requirement agents and fuzzy function agents, (3) building interaction between fuzzy function agents, (4) building interactions between fuzzy function agents and fuzzy solution agents, and (5) building interactions between fuzzy constraint agents and fuzzy solution agents.

In the second phase, the fuzzy set of consensual solution agents emerges. First the *fuzzy set of requirements* for a particular customer is defined. Given the fuzzy set of customer requirements, the fuzzy set of product function agents are computed using the fuzzy relationship between requirement agents and product function agents (*active functions)*. As soon as the set of active function agents emerge, solutions agents interact with them to compute the *fuzzy set of solutions*, (*active solutions)*. At the same time, constraints of different process views, involved in the configuration task, are defined. After integration of constraints, fuzzy constraint agents interact with active solution agents to converge towards fuzzy sets of solutions satisfying all fuzzy constraints. The fuzzy set of consensual solution agents emerges after the intersection of active fuzzy set of solution agents satisfying customer requirements, fuzzy set of solution agents satisfying all fuzzy constraints.

In the third phase, the optimal configuration emerges from fuzzy consensual solution agents and their affinities. During this phase, the consensual solution agents through their interactions and using their affinities are structured into modules. Maximization of interactions between the consensual solution agents within a module and minimization of interactions of consensual solution agents in-between modules is the objective function to be optimized.

In the fourth, final phase agents seek the consensus. Interactions between fuzzy solution agents and fuzzy configuration agents are built so that the consensus emerges. Thus, consensus agents interact with fuzzy solution agents as well as with the fuzzy configuration agents. They can inform the designer about the different coefficients established to measure the consensus that emerged. Among the advantages provided by discerning the consensus nuclei of configurations by fuzzy agents, is the creation of a common ground for moving toward an acceptable configuration and, thus, it provides assistance to the collaborative and distributed design for configuration. In addition, the fuzzy consensus can promote the capitalization of consensual configurations according to consensual functional requirements and consensual constraints. It permits the expertise of the various actors to be shared.

14.4.4 PDM Approach

In modern PDM systems, the overall structure of a modular product is mapped in a generalized product structure, the so-called "150-percent-bill-of-material" (BOM). Alternative or optional items are initially managed in the database of PDM systems in the same way as all other items, i.e., items as master records with corresponding attributes. Differences to the usual article management arise only in the structuring of the product in the form of bills. Through the use of variants in product structures, PDM systems are able to manage order neutral BOMs with varying and optional positions. This approach is beneficial for product development and less for production and accompanied departments, because there explicit BOMs are needed for each product variant to be produced.

Furthermore, there is a risk that the data management is very complicated, while compromising the performance of the system needs to be tolerated, especially when a large number of product variants needs to be managed. To resolve these conflicts, modern PDM systems are extended by the module "ariant Manager" of which a schematic workflow is depicted in Fig. 14.9. In the base module all master data (parts, structures and processes) are managed. In case of variants explicit ones are derived by the configuration and clone modules. Various reports can be generated by a reporting module that also contains neutral data when needed.

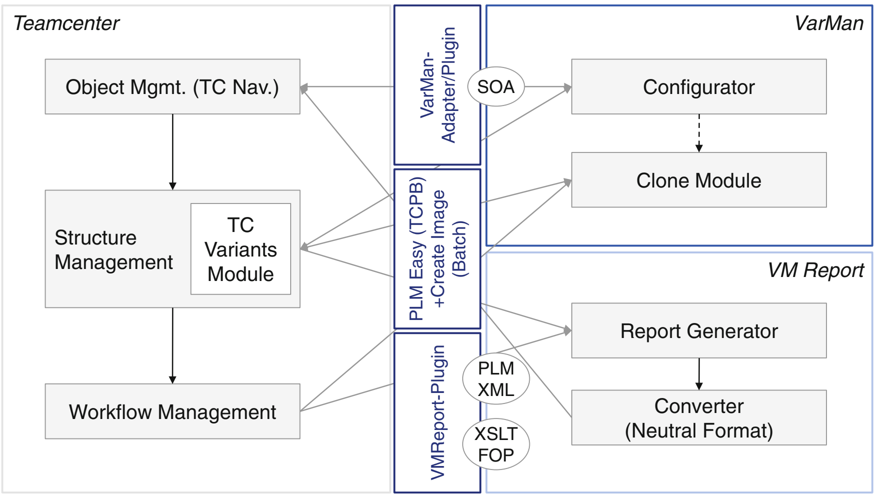


Fig. 14.9 Modular design driven by PDM system

14.5 Use Cases

Several examples of configurable products have been studied in the literature such as: cars, elevators, computer equipment, computer software, telephone switching systems, telecommunication networks, etc. Many companies such as Siemens [44], Mercedes [45], Jaguar Land Rover [46], Volvo Trucks and Fiat [47] have their own history in the development of configuration technologies and tools. This chapter is completed by three use cases, which demonstrate the support of modular design by appropriate configuration tools in the automotive and aerospace industry, and plant manufacturing.

14.5.1 Automotive

Jaguar land rover (JLR) is a premium vehicle manufacturer with two brands: Jaguar produces high performance sports cars and saloons, whilst Land Rover produces class leading 4-wheel drive vehicles.

Caused by JLR's varied ownership heritage two product-definition authoring configurators still exist, both originated in the early to mid-1990s. Both configurators can handle "f-Then"rules and they have a constraints capability. Three cooperating IT applications have been designed to enable the company to operate as an assemble-to-order manufacturer: a product definition application, a bill-of-material system and an order-scheduling system. However, configuration data is increasingly to be distributed across multiple applications throughout the enterprise. In the context of assemble-to-order configuration, a use case has demonstrated a range of configuration concepts to be modeled which are representative of the complexity of JLR's vehicles [46]. Based on these experiences JLR decided to introduce a single-rule authoring and configuration management application, which encompasses product configuration in an integral module. Table 14.2 summarises the configuration concepts and configuration principles.

It was found that neither a PLM, nor an ERP-oriented standard application is able to supply the needed functionality for a lifecycle approach to product configuration. PLM systems are product-centered tools, whereas ERP systems consists of operational business tools. A JLR's specific configuration lifecycle management (CLM) system was proposed to support the complete product lifecycle for a JLR configuration [47]. Table 14.3 shows some aspects of PLM, CLM and ERP and their differences from a product configuration management perspective.

The interfacing between PLM, CLM and ERP is conveniently built on product features and their families. Features are the basic building blocks for defining configurations of vehicles and provide the basis for one common modelling language through which to write rules (see Sect. 10.6.3). The features include customer-facing features used in the ordering of products as well as technical features, which are unimportant to customers but are essential in driving the manufacturing

Table 14.2 Configuration concepts and configuration principles [46]

Configuration concepts tested	Configuration principles
1. Support for effectivity	• Support for effectivity based configuration throughout the product lifecycle.
	• Effectivity is a time point or set of points which define the availability period of an object e.g. a feature, a feature rule, a feature specification, etc., in a product.
2. Support for concurrent build periods	• Differentiation between production builds and prototype builds using effectivity.
	• Releases managed around timing points and change points, with multiple programms of work being managed independently.
3. Support of an integrated modelling environment	• Support for both perpetual and model year based releasing in the product-modelling environment.
4. Support for rule inheritance	• Creation of marketed intent will be defined within the bounds defined by the engineered intent.
5. Support for the authoring cross-carline rules	• Enable modular product modelling, for example, homologation rules and certain architectural technical constraints apply accross multiple car-lines.

Table 14.3 Aspects of PLM, ERP and CLM from configuration management perspective [46]

	PLM	CLM	ERP
Purpose	Supporting product knowledge management.	Supporting configuration lifecycle management.	Supporting operational business requirements.
Approach	Project-based.	Supports both project based and transactional view with one set of configuration master data.	Transaction-based.
Time cycles	Supports time to market.	Supports both time to market and time to customer.	Supports time to customer.
BOM focus	Definition of an engineering BOM.	Generation of solved feature strings.	Execution of a manufacturing BOM.
Solve performance	BOM-explosion workload: 10's per hour.	Performs many types of solves also to support, among other things, BOM explosion at a rate of 10000 s per hour.	BOM-explosionwork-load: 1000 s per hour.
Types of constraints authored	Technical constraints.	Technical and commercial constraints coexist.	Commercial constraints.
Change of product configuration offerings	By model year.	Linking model years and running changes.	Running.
Configuration space	Partially defined configuration space biased towards technical features.	Fully defined configuration space linking technical and commercial features.	Partially defined configuration space biased towards commercial features.

processes. Thus, an important shared resource is a master feature dictionary, which contains the common vocabulary of all allowed features across the involved systems. A master feature dictionary should be managed carefully to avoid duplications of features. The master feature dictionary will evolve over time and some re-modelling of feature families is unavoidable as vehicle technologies develop and otherwise simple features change into combinations of sub-features.

14.5.2 Aerospace

An automated configuration design method and tool for the realistic representation of various aerospace vehicle geometries, using fewer control parameters, build an aircraft geometry step-by-step using typical and common components. It is intended for using them in conceptual and preliminary design phases [48].

This tool, called PCAD, is based on effective use of a super-elliptic formulation with exponential and polynomial distribution functions in an example of configuration design. With this design method, designers can represent the geometry either of airplane, helicopter, fighter or missile more realistically, quickly and efficiently. After that, they can store the geometry data in a PDM system, and manipulate complex shapes with a small number of control parameters. An aircraft geometry model in an associative environment, which is represented more realistically and precisely, is applicable to different stages of the design and development lifecycle, and can be edited like any other interactively created CAD model.

PCAD encompasses several modules: a graphical user interface (GUI), an aircraft “configuration scheme selection” decision-making tool, “surface coordinate points generator” (“control points grid”), customized CAD software adapted to the needs of designers and a commercial PDM tool and is fully embedded in CATIA/Enovia environment by using CATIA V5 CAA interfaces (Fig. 14.10).

14.5.3 Plant Design with Product Configurator and KBE

In plant design, machines with more than 10.000 parts are applied which are documented in 3D-CAD and PDM systems. They are customized by the following criteria: market and customer requirements, technical producibility, own business aims and the general possibility to create modules. Thereby, both arbitrary complexity and the reduction of product offering have to be avoided. The right product configuration is generated by a Web based product configurator. Additionally, a convenient product presentation for given configuration is chosen by using KBE.

Even for factory planning with more than 500 machines in one production hall and internet applications, complex models which show every detail cannot be used for performance reasons. Furthermore, no company wants to share its know-how with its competitors through the internet discovering fully detailed CAD models.

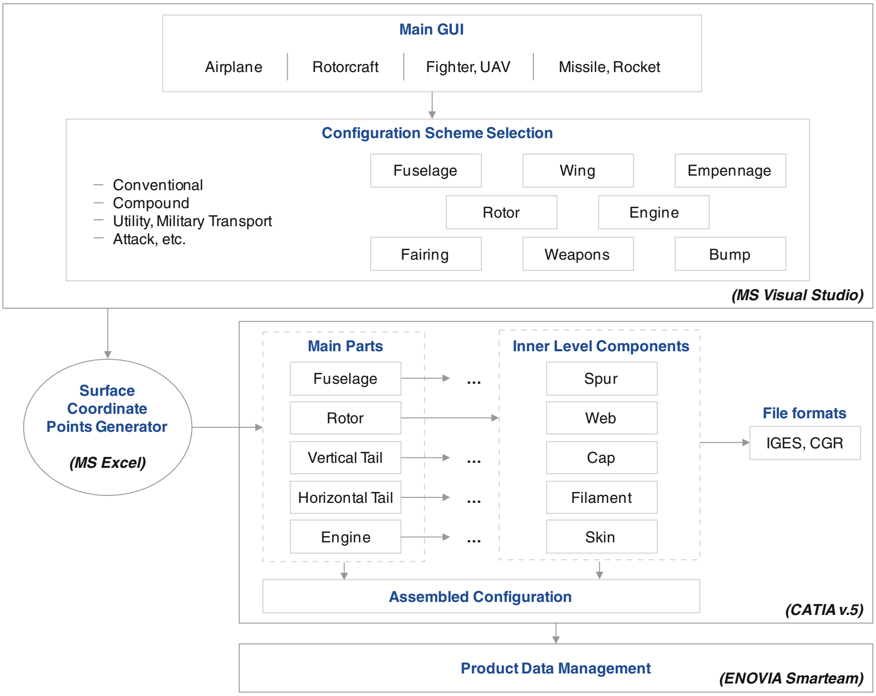


Fig. 14.10 Aircraft configuration design tool [48]

The key is the separation of complex and simplified CAD models in two different data sets, which, though, are managed by the same status information. The simplified model can be generated in different characteristic features (simple, mid, complex). As an example, the sales staff discusses the design of the machine hall with the client and configures the design in 3D on site. In doing so, the characteristics of the individual machines are written down in the CAD system and the simplified models are used. A prime scale drawing can be printed locally and an offer is generated directly. Moreover, it is possible to add a rendered 3D picture to the offer.

In the example of Fig. 14.11 the machine designed by KBE and CAD is able to adapt every of the 50 million possible combinations in the CAD model. According to the client´s choice, the desired variant is adjusted with the product configurator. This variant is checked for doublets at any level of the structure and checked automatically in the PDM system. The parts can be produced directly in the connected sheet bend machine. The variant selection can be effected by an internet solution with the simplified model and, hence, can be directly passed to order management. The processing time for one job is reduced from days to hours.

Configurator can communicate bidirectionally with other sections and internal systems (CRM, ERP, PDM) by detached status information from the CAD system.

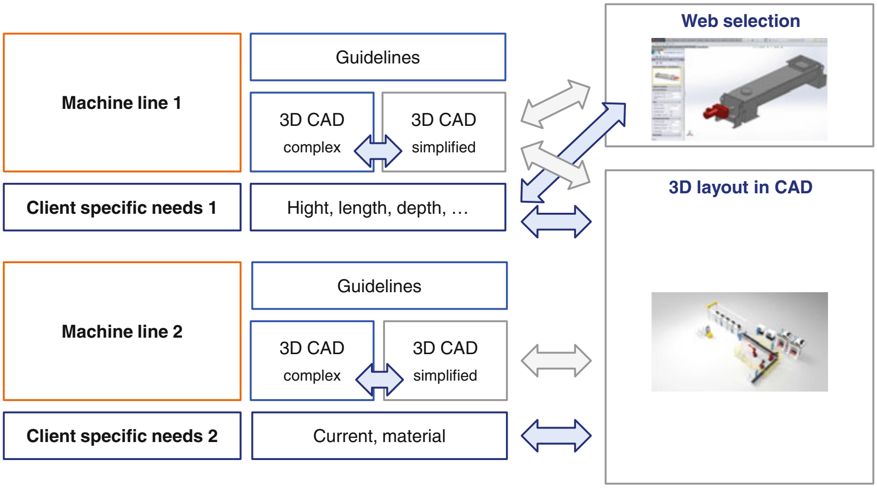


Fig. 14.11 Modular design with product configurator and KBE (Reproduced with kind permission from Lino GmbH © 2014 [49])

This solution allows building up bottom up relationship knowledge and setting up assembly plans by ERP object lists. Similar concepts, which combine product configuration with KBE are used for the design of automotive components [50].

14.6 Further Development

Design for product variety, design for product configuration, and design for mass customization are considered to be highly collaborative and distributed processes. *Therefore, from a holistic point of view, there is still much to be desired in order to achieve system-wide solutions for these design processes and platform-based product development, which can consider collaboration and distribution, intensive interaction between distributed actors, heterogeneity, dynamics and evolution of organization, and the uncertainty.*

The design of a modular product is considered to resolve a system-based interdependency problem. Traditionally, this issue has been seen as system architect's task. Architects design a functional and physical architecture of a system and their greatest concerns are still with the systems' connections and interfaces. The development of modular designs often requires a redesign of the components themselves resulting in new components. Consequently, an architect should assess the achievable technical performance of systems based on their underlying modular or integral architecture. *Modular design should be the result of a coherent and rationale design process, where the options, modular or integral, are early explored in response to technical constraints and the set of requirements. Finding* *the relationship between sparseness, modularity, technical constraints and the set of requirements, could allow such assessment early in design process.* A task in modularity assessment is also the issue of increasing the effectiveness of modularity. *Finding the relationship between the level modularity and the effectiveness of modularity is an open-ended issue.*

Product configuration and modularity are inherently related to product architecture. As the product architecture is considered to be the governing force in lifecycle design, the issue of product architecture lacks theoretical foundation. The design of product architecture has been considered rather more as a know-how issue of architects than a scientific-engineering issue. *In what ways a product architecture, accounting nowadays only for the functional and physical aspects of a modular product, integrate all other lifecycle characteristics is an important issue.*

Actually, the lifecycle of a module is confined to predefined scenarios that depend on its interfaces and its connections. A product with increased adaptability and suitability requires more efforts of design and manufacturing due to increased variety and complexity. *How to design intelligent modules is an important issue related to the design of intelligent products.*

The use of open architecture in modular design is a solution to allow the adoption of new technology. The use of existing modules as well as the use of independently developed modules to design new modular systems, while respecting the integrity of these modules, have to do with the suitability for integration of modules. The adaptability and suitability of modules for integration in a wide range of possibly larger systems is an important issue of the design and development of intelligent systems. *The concept of an intelligent product should maximize the design space of architects and system designers.*

The change management of requirements, functions, solutions and process constraints is another question in modular design. The development of intelligent modular products is strongly related to the development of intelligent models and intelligent tools. Thus, development of intelligent multidisciplinary collaborative and distributed platforms can better handle the modularity and variant management problem. *The multi-agent paradigm has the potential to respond to this challenge and to pave the way for the introduction of innovative technologies in a dynamic environment characterized by important changes and evolution.*

Development of intelligent models and intelligent tools on the one hand and the development of intelligent modular products, on the other, which can communicate and cooperate between them, need holistic and concurrent engineering approaches. *These approaches can offer the possibility of the design of self-sustainable models and self-sustainable products.*

To create long-lived modular systems, the foundations of the system have to reflect the corresponding relevant reality. The design of a modular product should exploit this principle thoroughly. More modularity is better in all lifecycle viewpoints. However, except architects, other actors like development project team members and management in general have often limited access to dependency-based system views. *Transfer and sharing of knowledge, from architect to various actors and vice versa, are essential to be able to support all lifecycle viewpoints in* *system level project coordination. If collaborative design in this context is to be successful, it must be built on a shared rationale of critical design decisions.*

A key motivation of modularity is the specialization in the design and production of modules. The modular product serves much larger user groups over longer periods of time than a single combined product. Thus, the performance of the structure of modular product reflects the performance of actors' coordination in an organization. Should a modular organization in a *dynamic world* reflect the modularity of the product, and, should a modular product reflect the modular organization, are still open questions. *Thus, finding the relationship between the performance of the structure of modular product and the performance of coordination of an organization could allow the early assessment of modular product design.*

14.7 Conclusions and Outlook

Modules are encapsulated groups of similar interconnected physical components that operate on a flow of energy, material, or information and perform a set of functional requirements. Autonomy or independence from external, dependence of internal is an important characteristic of modules. Minimization of interactions with external components and maximization of interactions between components within the module is an important characteristic of modules. The common attributes of modular products are: commonality of modules, combinability of modules, function binding in modules, interface standardization between modules, and higher interaction within module versus lower interaction in-between modules.

Modularity is an important property of product design as a multidisciplinary concept. In the context of concurrent engineering methods, modularity can be defined as the degree to which a product's architecture is composed of modules to respond to a set of requirements, including lifecycle issues and the organization of collaborative and distributed design processes. It includes also the organization of quantities of data, information, and knowledge of these design processes.

Design methods for modular design use principally the decomposition principle. The decomposition of functions into ordered sub-functions such that a group of ordered sub-functions can be modularized, the decomposition of relationship between components such that a cluster of components can define a module, as well as the decomposition of the relationship between the function and components such that a cluster of components corresponds to a cluster of functions, are the most used modular design approaches. Methods such as MFD use these principles from requirement to production. The goal is to facilitate the development of a "robust" production program adaptable to the varying requirements. Modularity is also used as a key measure of design for product variety, design for product configuration and design for mass customization. To optimize modularity the Axiomatic Design, the Design Structure Matrix, and the advanced variant mode and effects analysis (VMEA) can be used.

Various technologies and tools have been developed and used to achieve system-wide solutions for modular designs. The current trend is to use, combine and integrate different technologies such as advanced CAD systems, product configurators, agent based systems and PDM systems. Development of intelligent models and intelligent tools as well as the development of intelligent modular products (i.e. intelligent model-tool-product system), which can communicate and cooperate, demands the design of more intelligent organizations of designs processes for product variety, for product configuration, and for mass customization. Development of intelligent model-tool-product systems needs the development of holistic and concurrent engineering approaches. These approaches can offer the possibility of the design of intelligent self-sustainable models and intelligent self-sustainable products.

References

1. Manovich J (2001) The language of new media. MIT Press, Cambridge
2. Fodor JA (1983) The Modularity of Mind. MIT Press, Cambridge
3. Salvador F (2007) Toward a product system modularity construct: literature review and reconceptualization. IEEE Trans Eng Manag 54(2):219–240
4. Sosa ME, Eppinger SD, Rowles CM (2007) A network approach to define modularity of components in complex products. J Mech Des 129(11):1118–1129
5. Boysen N, Scholl A (2009) A general solution framework for component-commonality problems. Bus Res 2(1):86–106
6. Jiao JR, Simpson TW (2007) Siddique Z (2007) Product family design and platform-based product development: a state-of-the-art review. J Intell Manuf 18:5–29
7. Baldwin CY, Clark KB (2006) Modularity in the design of complex engineering systems. In: Braha D, Minai AA, Bar-Yam Y (eds) Complex engineered systems—science meets technology. Springer, Berlin, pp 175–205
8. Kuntz L, Vera A (2007) Modular organization and hospital performance. Health Serv Manag Res Royal Soc Med Press 20(1):48–58
9. Baldwin CY, Clark KB (1997) Managing in an age of modularity. Harvard Business Review 75(5): 84–93. Harvard Business School Publ. Boston
10. Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Engineering design a systematic approach, 3rd edn. Springer, London
11. Priest JW, Sanchez JM (2001) Product development and design for manufacturing: a collaborative approach to producibility and reliability, 2nd edn. Marcel Dekker, New York
12. Ehrlenspiel K, Kiewert A, Lindemann U (2007) Cost-efficient design. Springer, Berlin
13. Ong SK, Xu QL, Nee AYC (2008) Design reuse in product development modeling, analysis and optimization. World Scientific Publishing, Singapore
14. Gawer A (2009) Platforms, markets and innovation. Edward Elgar Publishing, Cheltenham
15. PillerFT Tseng MM (2010) Handbook of research in mass customization and personalization. World Scientific Publishing, Singapore
16. Fogliatto FS, da Silveira GJC (2011) Mass customization: engineering and managing global operations. Springer, London
17. Parry G, Graves A (2008) Build to order: the road to the 5-day car. Springer, London
18. Hvam L, MortensenNH Riis J (2008) Product customization. Springer, London
19. Hüttenrauch M, Baum T (2008) Effiziente Vielfalt Die dritte Revolution in der Automobilindustrie. Springer, Berlin

20. Junge M (2005) Controlling modularer Produktfamilien in der Automobilindustrie. Deutscher Universitätsverlag, Wiesbaden
21. Wei G, Qin Y (2012) Framework of rapid product cost estimation based on the modular product family. In: Chen R (ed) Proceedings of 2011 international conference in electrics communication and automatic control. Springer, London, pp 9–14
22. Rapp T (2010) Produktstrukturierung: Komplexitätsmanagement durch modulare Produktstrukturen und –plattformen, 2nd edn. Books On Demand, Norderstedt
23. Schuh G, Arnoscht J (2012) Aleksic S (2012) Systematische Gestaltung von Kommunalitäten in Produkten und Prozessen. ZFW, Jahrg 107(5):322–326
24. Daniilidis C, Enßlin V, EbenK, Lindemann U (2011) A classification framework for product modularization methods. ZFW, Proceedings of the 18t[h] international conference on engineering design, ICED11
25. Suh NP (2001) Axiomatic design: advances and applications. Oxford University Press, Oxford
26. Holmberg G (2002) A modular approach to the aircraft product development capability. In: 23r[d] congress of international council of the aeronautical sciences, 8–13 Sep, 2002, Toronto. http://www.icas.org/icas_archive/icas2002/papers/652.pdf. Accessed 15 July 2013
27. Hirtz J, Stone RB, McAdams DA, Szykman S, Wood KL (2002) A functional basis for engineering design: reconciling and evolving previous efforts. Res Eng Des 13:65–82
28. Stone RB, Wood KL (2000) A functional basis for engineering design: reconciling and evolving previous efforts. Des Stud 21(1):5–31
29. Eppinger SD, Browning TR (2012) Design structure matrix methods and applications. MIT Press, Cambridge
30. DSMweb.org. http://www.dsmweb.org/en/dsm.html. Accessed 15 July 2013
31. Cheng Q, Zhang G, Gu P, Shao X (2012) A product module identification approach based on axiomatic design and design structure matrix. Concurrent Eng 20:185
32. Erixon G (1998) Modular function deployment–a method for product modularisation, PhD thesis, The Royal Institute of Technology, Stockholm
33. Bergman B, de Mare J, Loren S, Svensson T (2009) Robust design methodology for reliability: exploring the effects of variation and uncertainty. Wiley, Chichester
34. Fischer JO (2008) Kostenbewusstes Konstruieren: Praxisbewährte Methoden und Informationssysteme für den Konstruktionsprozess. Springer, Berlin
35. Tiihonen J et al (1996) State of 10 cases in the Finnish industry. In: Tomiyama T, Mäntylä M, Finger S (eds) Knowledge intensive CAD. Chapman & Hall, London, pp 95–114
36. Brinkop A (2013) Marktführer Produktkonfiguration, Version 1.25, 4. Juli 2013. http://brinkop-consulting.com/guide/marktfuehrer.pdf. Accessed 15 Oct 2013
37. Ostrosi E, Fougères A-J, Ferney M (2012) Fuzzy Agents for product configuration in collaborative and distributed design process. Appl Soft Comput 12(8):2091–2105
38. Deciu ER, Ostrosi E, Ferney M, Gheorghe M (2005) Configurable product design using multiple fuzzy models. J Eng Des 16(2–3):209–235
39. Munoz-Hernandez S, Gomez-Perez JM (2005) Solving collaborative fuzzy agents problems with CLP (FD). LNCS 3350:187–202
40. Ostrosi E, Fougères AJ (2011) Optimization of product configuration assisted by fuzzy agents. Int J Interact Des Manuf 5(1):29–44
41. Ostrosi E, Haxhiaj L, Ferney M (2008) Configuration grammars: powerful tools for product modelling in cad systems. In: Curran R et al (eds) Collaborative product and service life cycle management for a sustainable world, Proceedings of the 15t[h] ISPE international conference on concurrent engineering (CE 2008). Springer, London, pp 451–459
42. Ostrosi E, Fougères AJ, Ferney M, Klein D (2012) A fuzzy configuration multi-agent approach for product family modelling in conceptual design. J Intell Manuf 23(6):2565–2586
43. Fougères AJ, Ostrosi E (2013) Fuzzy agent-based approach for consensual design synthesis in product configuration. Int Comput Aided Eng 20:259–274
44. Falkner A, Haselböck A (2012) An overview of configurator use at Siemens, 2012 Oxford Configuration Workshop, 12–13 Jan, Oxford University. http://www.cs.ox.ac.uk/OxfordConfiguration/. Accessed 15 Oct 2013

45. Kübler AJ, Zengler C (2010) Model counting in product configuration. Workshop on logics for component configuration (LoCoCo 2010) EPTCS 29, pp 44–53
46. Batchelor J, Andersen HR (2012) Bridging the product configuration gap between PLM and ERP—an automotive case study. In: 19th international product development management conference, Manchester, 17–19 June 2012
47. Helo PT, Xu QL, Kyllonen SJ, Jiao RJ (2010) Integrated vehicle configuration system—connecting the domains of mass customization. Comput Ind 61(1):44–52
48. Azamatov A, Lee JW, Byun YH (2011) Comprehensive aircraft configuration design tool for integrated product and process development. Adv Eng Softw 42(2011):35–49
49. Lino GmbH. http://www.lino.de/3d_layout.html. Accessed 15 Feb 2014
50. Elgh F (2014) Automated engineer-to-order systems A task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3–4):324–347

Chapter 15
Multidisciplinary Design Optimization: Designed by Computer

Cees Bil

Abstract Multidisciplinary design optimization (MDO) has been a field of research for 25 years. It refers to the formulation of the design problem in mathematical models and applying optimization techniques to find the minimum or maximum of a predefined objective function, possibly subject to a set of constraints. MDO has become an important tool in concurrent engineering (CE), with the ability to handle many design variables (DV) across various disciplines. Advances in computer technologies and software engineering have facilitated the practical application of MDO in industry, including aerospace, automotive, shipbuilding, etc. However, active research and development in MDO continues. The creative input of the human designer to the design process is critical and must be integrated in the MDO process. For MDO to be effective in the design of modern complex systems it must also incorporate non-technical disciplines, such as finance, environment, operational support, etc. It remains a challenge to do model them with adequate fidelity. Simulations and analytical models have imbedded assumptions, inaccuracies and approximations. How do we deal with these in an MDO environment? This chapter gives an introduction to MDO with an historical review, a discussion on available numerical optimization methods each with their specific features, the various MDO architectures and decompositions and two case studies where MDO has been applied successfully.

Keywords Multidiscplinary design optimization · Complex systems · Concurrent engineering · MDO architectures · Optimization methods

C. Bil (✉)
RMIT University, GPO Box 2476, Melbourne, VIC 3001, Australia
e-mail: cees.bil@rmit.edu.au

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_15

15.1 Introduction

The whole is more than the sum of its parts—Aristotle

When the Wright brothers made their historical flight in 1903, their objective was to achieve powered and controlled flight. In the twenty first century, achieving powered and controlled flight is hardly the challenge anymore, the question is how well will it fly and will it meet the user's needs. The user's needs are not necessarily focused on hardware, but on a total business solution, including maintenance, support, upgrades, etc., that achieve a certain objective over the life-cycle of the system. Since the industrial revolution, engineers have invested their ingenuity in developing increasingly complex machines, but perhaps the most striking development in terms of rapid technical progression and complexity is the aerospace domain (Fig. 15.1).

The current design environment of complex systems is defined by a rapid turnaround of cost effective solutions, involving all operational and business aspects. The concurrent engineering (CE) approach considers all technical and business aspects simultaneously, rather than sequentially as in the traditional design approach. A sequential design approach does not guarantee that an overall optimum design is found. Figure 15.2 shows a typical aircraft design problem. In the sequential design process, the aerodynamics group determines the best aspect ratio (AR), for example, for maximum range P, subject to performance requirements (design 01). Unfortunately, the structures group cannot comply with the flutter requirement and needs to increase the wing weight W_{min} (design 02). For design 02 all requirements are met, but it is not the optimum design (design 03). Considering

Fig. 15.1 Evolution of engineering complexity in the past century

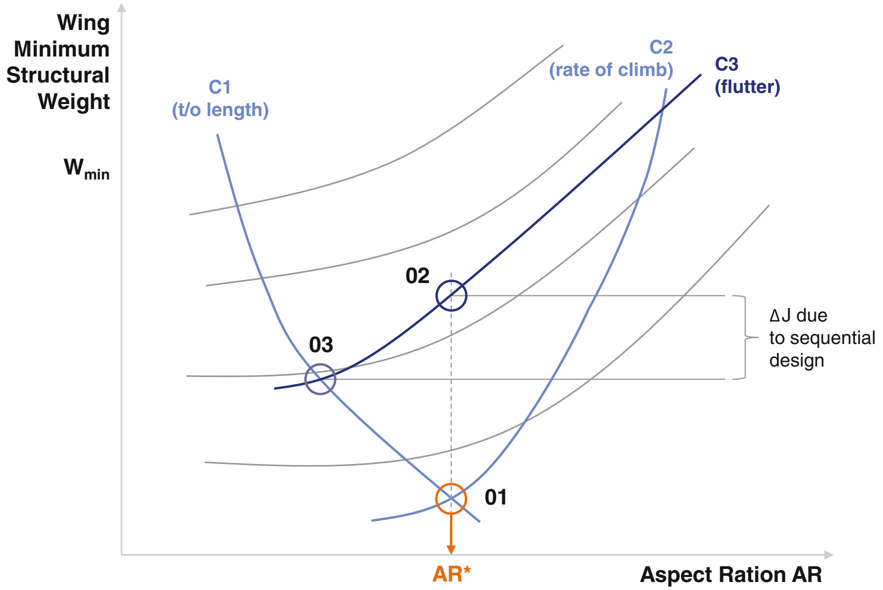


Fig. 15.2 Sequential versus concurrent design process [1]

aerodynamics, structures and performance at the same time, i.e. concurrently, would have resulted in an improved design.

This chapter introduces the multidisciplinary design optimization (MDO) approach that represents a modelling and simulation environment where numerical optimization techniques are applied to drive the optimization process. This chapter gives an overview of MDO applications to complex systems design. Section 15.2 provides the motivation for using MDO and its potential benefits in reduced lead times and improved design quality. Section 15.3 gives an historical background to MDO development. Section 15.4 discusses a range of numerical optimization methodologies, their classification and specific features. Section 15.5 cover more specifically nonlinear optimization methodologies, including the gradient-based methods such as SQL and GRM, and the genetic algorithms (GA) which have gained recent popularity as they do not rely on gradient information and are able to find a global optimum. Section 15.6 discusses MDO techniques for cases which are multi-modal or have multiple objectives. Section 15.7 gives an overview of various MDO architectures and the opportunity to decompose the optimization problem into different levels and coupling of variables, which avoids redundant computations and can speed up the process considerably. Section 15.8 presents two case studies where MDO was applied successfully in the structural design of a car body and of an aircraft wing. Section 15.9 concludes with a discussion some of the impediments in MDO application and focus areas for future research and development.

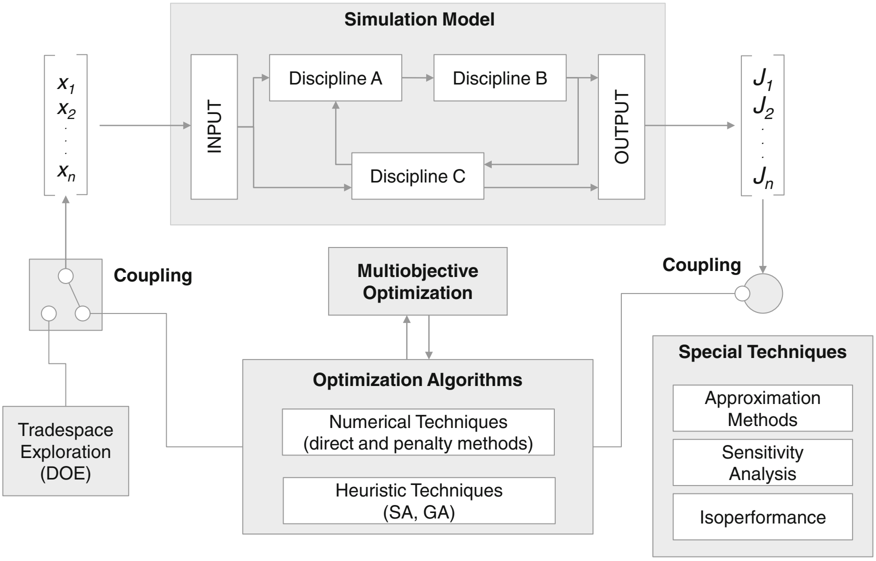


Fig. 15.3 Generic MDO framework [1]

15.2 Multidisciplinary Design Optimization (MDO)

A new approach that has gained much interest in the past two decades in assisting design teams is MDO [2, 3]. MDO is a sub-field of computational engineering and proposes an environment where all the relevant analysis tools, or simulation models, are coupled and a numerical optimization algorithm is applied to search for the optimum design as defined by a given objective function and subject to design constraints (Fig. 15.3).

There are a number of advantages to the MDO approach, such as:

- Reduction in design time
- Systematic, logical design procedure
- Handles wide variety of design variables (DV) and constraints concurrently
- Not biased by prejudice

These potential benefits have motivated many researchers, scientists and engineers to develop MDO frameworks for a range of different application [4–10].

15.3 Historical Background

The existence of optimization methods is as old as calculus and can be traced to the days of Newton, Lagrange and Cauchy [11]. The development of differential calculus methods of optimization was possible because of the contributions of

Newton and Leibnitz. The foundations of calculus of variations were laid by Bernoulli, Euler, Lagrange and Weierstrass. The optimization of constrained problems, which involves the addition of unknown multipliers, became known by the name of its inventor Lagrange. Cauchy made the first application of the steepest descent method to solve unconstrained minimization problems. In spite of these early contributions, very little progress was made until the middle of the 20th century, when high-speed digital computers made the implementation of the optimization procedures possible and stimulated further research in new methods.

The first step in the application of optimization was in structural design in the 1960s when Schmit [12] proposed a rather general new approach, which served as the conceptual foundation for the development of many modern structural optimization methods. It introduced the idea and indicated the feasibility of coupling finite element structural analysis and non-linear mathematical programming to create automated optimum design capabilities for a rather broad class of structural design problems.

An alternative, analytical form of structural optimization was offered by Prager and in numerical form by Venkayya in 1968 [13]. This concept became known as the optimality criteria. In the design of statically determinate structures, each member is fully stressed under at least one loading condition. The strength of the two methods suggested a natural separation of the design problem, where optimality criteria would deal with a large number of DV and mathematical programming would solve the component-design problem. This approach was pursued by Sobieski et al. in 1972 in the design of fuselage structures.

For practical MDO applications there are two important issues. The first is the selection of the models and analysis methods. As mathematical optimization relies only on the analysis methods provided; these methods must not only be accurate, but also correctly reflect the sensitivity to variations in the selected DV. The choice of analysis methods will depend on the design phase. It is usually not appropriate to use a Navier-Stokes CFD code in conceptual design as design is still very flexible and not accurately defined yet. Instead statistical/empirical methods as found in are more appropriate in the early design stages. A number of computer-based design synthesis systems have been developed for aircraft configuration design, such as ACSYNT, ADAS, RDS, SOCCER and AAA. Note that statistical/empirical methods are not based on engineering science and are therefore only applicable in a narrow range of applications and are not necessarily correctly sensitive to the selected DV.

The second issue is an acceptable computing time required to determine the optimum solution. This depends on the available computing power, sophistication of the analysis methods and the efficiency of the optimization method its and implementation. Investigations into using approximation methods as a mechanism to improve the efficiency of mathematical programming techniques started in the 1970s. This hybrid method uses approximations to find the optimum solution and then applies a more sophisticated analysis method to the approximate optimum design. The final optimum design is obtained iteratively. A form of this approach is known as surrogate models or response function techniques [14, 15].

15.4 Numerical Optimization Methods

Optimization is an important tool in decision science and in the analysis of physical systems. To use this methodology, we must first identify an objective, a quantitative measure of the quality of the system, for example profit, time, potential energy, or any quantity or combination of quantities that can be represented by a single numeric. The objective depends on certain characteristics of the system, called DV. The aim is to find the values for the DV that maximizes and minimizes the objective function. Often the range of values for the variables is constrained. The process of defining the relationship between the objective function, DV, and constraints for a given problem is known as modeling. Construction of an appropriate model is the first step—sometimes the most important step—in an optimization process. If the model is too simple, it will not give useful insights into the practical problem. If it is too complex, it may be too difficult to solve.

Once the model has been formulated, an optimization algorithm can be used to find its numerical solution. A variety of optimization algorithms exists, each tailored to a particular type of optimization problem. The responsibility of choosing the algorithm that is appropriate for a specific application often falls on the user. This choice is an important one, as it may determine whether the problem is solved rapidly or slowly and, indeed, whether the solution is found at all. After the optimization process has been completed, we must be able to recognize whether it has succeeded in its task of finding an optimum solution. In many cases, there are elegant mathematical expressions known as optimality conditions for checking that the current set of values for the DV is indeed the optimum solution of the problem. If the optimality conditions are not satisfied, they may still give useful information on how the current estimate of the solution can be improved. The model may be improved by applying techniques such as sensitivity analysis, which reveals the sensitivity of the solution to changes in the model and data. Interpretation of the solution may also suggest ways in which the model can be refined or improved (or corrected). If any changes are made to the model, the optimization problem is solved anew, and the process repeats.

15.4.1 Mathematical Formulation

In a mathematical context, optimization is the minimization or maximization of a function subject to constraints on its variables [16, 17]. We use the following notation:

- x is the vector of variables, also called unknowns or parameters;
- f is the objective function, a (scalar) function of x to be maximized or minimized;
- c_i are constraint functions, which are scalar functions of x that define certain equalities and inequalities that the unknown vector x must satisfy.

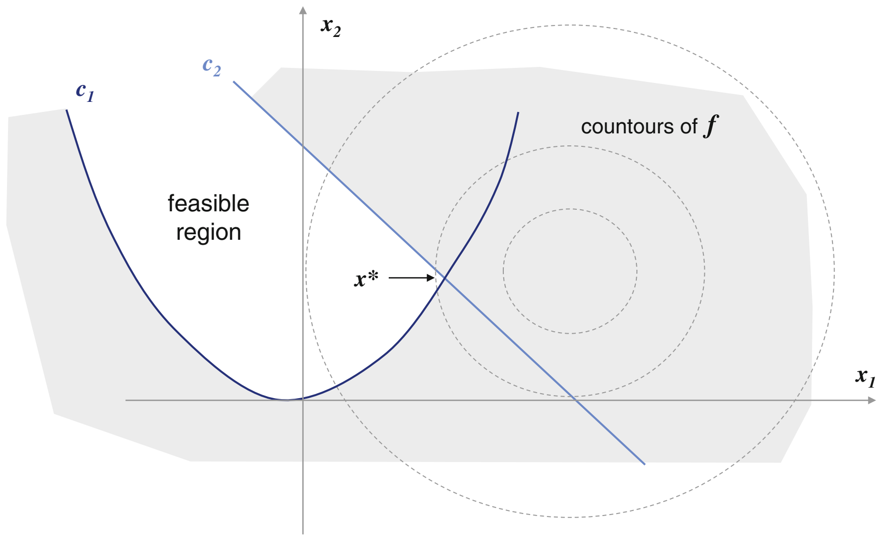


Fig. 15.4 Optimization problem with two design variables [18]

Using this notation, the optimization problem can be written as follows:

$$\min_{x \in R^n} f(x), \quad \text{subject to} \begin{matrix} c_i(x) = 0, i \in n_e \\ c_i(x) \geq 0, i \in n_i \end{matrix} \tag{15.1}$$

Figure 15.4 shows the contours of the objective function, that is, the set of points for which $f(x)$ has a constant value [18]. It also illustrates the feasible region, which is the set of points satisfying all the constraints (the area between the two constraint boundaries), and the point $x*$, which is the solution of the problem. The "infeasible side" of the inequality constraints is shaded. Classification of the engineering design optimization problem is necessary to select the right approach for a given problem [18, 19]. A classification is presented in Fig. 15.5. In the next sections different categories of optimization methods are discussed with their specific features and capabilities.

15.4.2 Constrained and Unconstrained Optimization

Problems with the general form of Eq. (15.1) can be classified according to the nature of the objective function and constraints (linear, nonlinear, convex), the number of variables, large or small, the smoothness of the functions, differentiable or non-differentiable, and so on. An important distinction is between problems that have constraints on the variables and those that do not. Unconstrained optimization problems, for which we have $n_e = n_i = 0$ in Eq. (15.1), arise in many practical

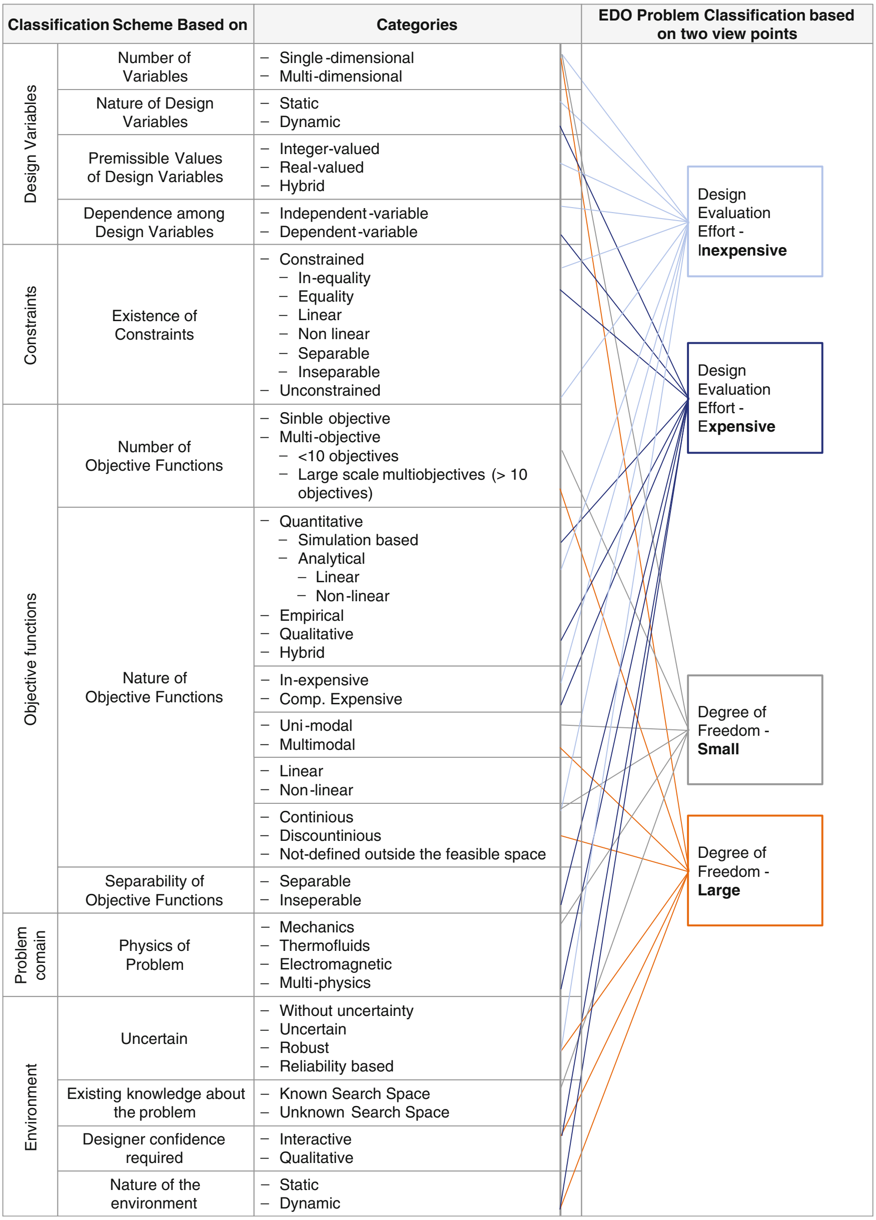


Fig. 15.5 Classification of optimization methods [18]

applications. Even for some problems with natural constraints on the variables, it may be appropriate to disregard them if they do not affect the solution and do not interfere with the algorithms. Unconstrained problems arise also as reformulations of constrained optimization problems, in which the constraints are replaced by penalization terms added to objective function that have the effect of discouraging constraint violations. Constrained optimization problems arise from models in which constraints play an essential role, for example in imposing budgetary constraints in an economic problem or shape constraints in a design problem. These constraints may be simple bounds, more general linear constraints, or nonlinear inequalities that represent complex relationships among the variables.

When the objective function and all the constraints are linear functions of x, the problem is a linear programming problem. Problems of this type are probably the most widely formulated and solved of all optimization problems, particularly in management, financial, and economic applications. Nonlinear programming problems, in which at least some of the constraints or the objective function is nonlinear, tend to arise naturally in the physical sciences and engineering, and are becoming more widely used in management and economic sciences as well [20, 21].

15.4.3 Continuous Versus Discrete Optimization

In some optimization problems the variables make sense only if they take on integer values. For example, a variable x could represent the number of power plants that should be constructed by an electricity provider during the next 5 years, or it could indicate whether or not a particular factory should be located in a particular city. The mathematical formulation of such problems includes integrality constraints or binary constraints, in addition to algebraic constraints like those appearing in Eq. (15.1). Problems of this type are called integer programming problems. If some of the variables in the problem are not restricted to be integer or binary variables, they are called mixed integer programming (MIP) problems. Integer programming problems are a type of a discrete optimization problem. Generally, discrete optimization problems may contain not only integers and binary variables, but also more abstract variable objects such as permutations of an ordered set. The defining feature of a discrete optimization problem is that the unknown x is drawn from a finite, but often very large, set. By contrast, the feasible set for continuous optimization problems is usually infinite, as when the components of x are allowed to be real numbers.

Continuous optimization problems are usually easier to solve because the smoothness of the functions makes it possible to use objective and constraint information at a particular point x to deduce information about the function's behavior at all points close to x. In discrete problems, by contrast, the behavior of the objective and constraints may change significantly as we move from one feasible point to another, even if the two points are "close" by some measure. The feasible sets for discrete optimization problems can be thought of as exhibiting an extreme form of non-convexity, as a convex combination of two feasible points is in general not feasible. Continuous optimization techniques often play an important role in solving discrete optimization problems. For instance, the branch-and-bound method for integer linear programming problems requires the repeated solution of linear programming "relaxations," in which some of the integer variables are fixed at integer values, while for other integer variables the integrality constraints are temporarily ignored. These sub-problems are usually solved by the simplex method.

15.4.4 Global and Local Optimization

Many algorithms for nonlinear optimization problems seek only a local solution, a point at which the objective function is smaller than at all other feasible nearby points. They do not always find the global solution, which is the point with lowest function value among all feasible points. Global solutions are needed in some applications, but for many problems they are difficult to recognize and even more difficult to locate. For convex programming problems, and more particularly for linear programs, local solutions are also global solutions. General nonlinear problems, both constrained and unconstrained, may possess local solutions that are not global solutions.

15.4.5 Stochastic and Deterministic Optimization

In some optimization problems, the model cannot be fully specified because it depends on quantities that are unknown at the time of formulation. This characteristic is shared by many economic and financial planning models, which may depend for example on future interest rates, future demands for a product, or future commodity prices, but uncertainty can arise naturally in almost any type of application.

Rather than just use a "best guess" for the uncertain quantities, more useful solutions may be obtained by incorporating additional knowledge about these quantities into the model. For example, they may know a number of possible scenarios for the uncertain demand, along with estimates of the probabilities of each scenario. Stochastic optimization algorithms use these quantifications of the uncertainty to produce solutions that optimize the expected performance of the model. Related paradigms for dealing with uncertain data in the model include chance constrained optimization, in which we ensure that the variables x satisfy the given constraints to some specified probability, and robust optimization, in which certain constraints are required to hold for all possible values of the uncertain data.

Many algorithms for stochastic optimization do, however, proceed by formulating one or more deterministic sub-problems, each of which can be solved by the aforementioned techniques. Stochastic and robust optimization are seeing a great deal of recent research activity.

15.4.6 Convexity

The concept of convexity is fundamental in optimization. Many practical problems possess this property, which generally makes them easier to solve both in theory and practice. If the objective function in the optimization problem (1) and the feasible region are both convex, then any local solution of the problem is in fact a global solution. The term convex programming is used to describe a special case of the general constrained optimization problem in which:

- Objective function is convex,
- Equality constraint functions ci $(\cdot)$, $i \in E$, are linear, and
- Inequality constraint functions ci $(\cdot)$, $i \in I$, are concave.

Optimization algorithms are iterative: they begin with an initial guess of the variable x and generate a sequence of improved estimates (called "iterates") until they terminate, hopefully at a solution. The strategy used to move from one iterate to the next distinguishes one algorithm from another. Most strategies make use of the values of the objective function f, the constraint functions c_i, and possibly the first and second derivatives of these functions.

Some algorithms accumulate information gathered at previous iterations, while others use only local information obtained at the current point. Regardless of these specifics, good algorithms should possess the following properties:

- Robustness: they should perform well on a wide variety of problems in their class, for all reasonable values of the starting point.
- Efficiency: they should not require excessive computer time or storage.
- Accuracy: they should be able to identify a solution with precision, without being overly sensitive to errors in the data or to the arithmetic rounding errors that occur when the algorithm is implemented on a computer.

These goals may conflict. For example, a rapidly convergent method for a large unconstrained nonlinear problem may require too much computer memory. On the other hand, a robust method may also be the slowest. Tradeoffs between convergence rate and storage requirements, and between robustness and speed, and so on, are central issues in numerical optimization.

The mathematical theory of optimization is used both to characterize optimal points and to provide the basis for most algorithms. It is not possible to have a good understanding of numerical optimization without a firm grasp of the supporting theory. Accordingly, this chapter gives a solid, though not comprehensive, treatment of optimality conditions, as well as convergence analysis that reveals the strengths and weaknesses of some of the most important algorithms.

15.5 Nonlinear Programming Techniques

Most MDO systems for complex engineering design will have to assume the general case that at least the objective function or one of the constraint functions are nonlinear. In that case a nonlinear optimization technique is used. In this category there are gradient-based methods that rely on first and second derivatives of the objective function and constraint functions to determine the search direction and update the DV. If they cannot be calculated implicitly, these derivatives can be approximated using a finite-difference method. Stochastic methods such as GAs do not require gradients and have therefore gained significant interest over the past few years.

15.5.1 Sequential Quadratic Programming

Sequential quadratic programming (SQP) methods are iterative methods that solve at the *k*th iteration a quadratic sub-problem (QP) of the form QP [22, 23]:

$$\text{Minimise}: \quad \min d^t H_k d + \nabla f(x_k)^t d \tag{15.2}$$

subject to

$$\begin{aligned} \nabla h_i(x_k)^t d + h_i(x_k) = 0, \quad i = 1, \ldots, p, \\ \nabla g_j(x_k)^t d + g_j(x_k) \leq 0, \quad j = p+1, \ldots, q \end{aligned}$$

where d is the search direction and H_k is a positive definite approximation to the Hessian matrix of Lagrangian function of problem (P). The Lagrangian function is given by:

$$L(x, u, v) = f(x) + \sum_{i=1}^{p} u_i h_i(x) + \sum_{j=p+1}^{q} v_j g_j(x) \tag{15.3}$$

where u_i and v_j are the Lagrangian multipliers. The sub-problem (QP) can be solved by using the active set strategy. The solution d_k is used to generate a new iterate:

$$x_{k+1} = x_k + \alpha_k d_k \tag{15.4}$$

where the step-length parameter $\alpha_k \in (0,1]$ depends on some line search techniques. At each iteration, the matrix H_k is updated according to any of the quasi-Newton method. The most preferable method to update H_k is Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, where H_k is initially set to the identity matrix I and updated using the equation:

$$H_{k+1} = H_k + \frac{y_k y_k^t}{s_k y_k^t} - \frac{H_k s_k s_k^t H_k}{s_k^t H_k s_k} \tag{15.5}$$

where

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla L(x_{k+1}, u_{k+1}, v_{k+1}) - \nabla L(x_k, u_k, v_k) \tag{15.6}$$

15.5.2 Generalized Reduced Gradient

The generalized reduced gradient (GRG) transforms inequality constraints into equality constraints by introducing slack variables [24]. Hence all the constraints in (P) are of equality form and can be represented as follows:

$$h_i(x) = 0, \quad i = 1, \ldots, q \tag{15.7}$$

where x contains both original variables and slacks. Variables are divided into dependent, x_D, and independent, x_I, variables (or basic and nonbasic, resp.):

$$x = \begin{bmatrix} x_D \\ \cdots \\ x_I \end{bmatrix} \tag{15.8}$$

The names of basic and nonbasic variables are from linear programming. Similarly, the gradient of the objective function bounds and the Jacobian matrix J may be partitioned as follows:

$$\begin{gathered} a = \begin{bmatrix} a_D \\ \cdots \\ a_I \end{bmatrix}, \quad b = \begin{bmatrix} b_D \\ \cdots \\ b_I \end{bmatrix}, \quad \nabla f(x) = \begin{bmatrix} \nabla_D f(x) \\ \cdots \\ \nabla_I f(x) \end{bmatrix} \\ J(x) = \begin{bmatrix} \nabla_D h_1(x) & \cdots & \nabla_I h_1(x) \\ \cdots & \cdots & \cdots \\ \nabla_D h_q(x) & \cdots & \nabla_I h_q(x) \end{bmatrix} \end{gathered} \tag{15.9}$$

Let x^0 be an initial feasible solution, which satisfies equality constraints and bound constraints. Note that basic variables must be selected so that $J_D(x^0)$ is nonsingular. The reduced gradient vector is determined as follows:

$$g_I = \nabla_I f(x^0) - \nabla_D f(x^0)(J_D(x^0))^{-1} J_I(x^0) \tag{15.10}$$

The search directions for the independent and the dependent variables are given by:

$$d_j = \begin{cases} 0, & \text{if } x_i^0 = a_i,\ g_i > 0 \\ 0, & \text{if } x_i^0 = b_i,\ g_i < 0 \\ -g_i, & \text{otherwise} \end{cases} \tag{15.11}$$

$$d_D = -(J_D(x^0))^{-1} J_I(x^0) d_t \tag{15.12}$$

A line search is performed to find the step length $f¿$ as the solution to the following problem:

$$\min f\left(x^0 + \alpha\, d\right) \tag{15.13}$$

Subject to: $0 \le \alpha \le \alpha_{\max}$, where $\alpha_{\max} = \sup\left\{\frac{\alpha}{a} \le x^0 \le x^0 + \alpha d \le b\right\}$.

The optimal solution α^* to the problem gives the next solution: $x^1 = x^0 + \alpha \cdot d$.

15.5.3 Genetic Algorithms

In the computer science field of artificial intelligence, a GA, evolutionary algorithms (EA) and particle swarm optimization (PSO) include a search heuristic that mimics the process of natural selection (biology-mimicking) [25–29]. This heuristic, also sometimes called a meta-heuristic, is routinely used to generate useful solutions to optimization and search problems [30]. GAs belong to the larger class of EA, which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. GAs are stochastic optimization algorithms based upon the principles of evolution observed in nature. Because of their power and ease of implementation, the use of GAs has noticeably increased in recent years. Unlike the gradient methods, they have no requirements on convexity, differentiability, and continuity of the objective, and constraint functions. These significant characteristics of GAs increase their popularity in applications. The basic GA can be summarized by the following steps:

1. Generate an initial population of possible solution (chromosomes) randomly,
2. Evaluate the fitness of each chromosome in the initial population,
3. Select chromosomes that will have their information passed on to the next generation,
4. Cross over the selected chromosomes to produce new offspring chromosomes,
5. Mutate the genes of the offspring chromosomes,
6. Repeat steps (3) through (5) until a new population of chromosomes is created,
7. Evaluate each of the chromosomes in the new population,
8. Go back to step (3) unless some predefined termination condition is satisfied.

GAs are directly applicable only to unconstrained problems. In the application of GAs to constrained nonlinear programming problems, chromosomes in the initial population or those generated by genetic operators during the evolutionary process generally violate the constraints, resulting in infeasible chromosomes. During the past few years, several methods were proposed for handling constraints, grouped into the following four categories:

- Preserving feasibility of solutions,
- Penalty functions,
- Search for feasible solutions,
- Hybrid methods.

Penalty function methods are the most popular methods used in the GAs for constrained optimization problems. These methods transform a constrained problem into an unconstrained one by penalizing infeasible solutions. Penalty is imposed by adding to the objective function $f(x)$ a positive quantity to reduce fitness values of such infeasible solutions:

$$\hat{f}(x) = \begin{cases} f(x) & \text{if } x \in F \\ f(x) + p(x) & \text{otherwise} \end{cases} \tag{15.14}$$

where $\hat{f}(x)$ is the fitness function and $p(x)$ is the penalty function whose value is positive. The design of the penalty function $p(x)$ is the main difficulty of penalty function methods. Several forms of penalty functions are available in the literature.

15.6 Multi-modal and Multi-objective Design Optimization

Optimization problems are often multi-modal: they possess multiple good solutions. They could all be globally good (same cost function value) or there could be a mix of globally good and locally good solutions. Obtaining all (or at least some of) the multiple solutions is the goal of a multi-modal optimizer [31–39].

Classical optimization techniques due to their iterative approach do not perform satisfactorily when they are used to obtain multiple solutions, since it is not guaranteed that different solutions will be obtained even with different starting points in multiple runs of the algorithm. EA however are very popular approaches to obtain multiple solutions in a multi-modal optimization task.

Real life engineering designs often have more than one conflicting objective functions thus requiring a multi-objective optimization approach. The multi-objective optimization becomes more difficult with increasing number of objectives and it has been shown in that existing multi-objective optimization algorithms do not perform well with more than five objectives. The optimization identifies several solutions that are good considering the objective functions. These are called Pareto solutions.

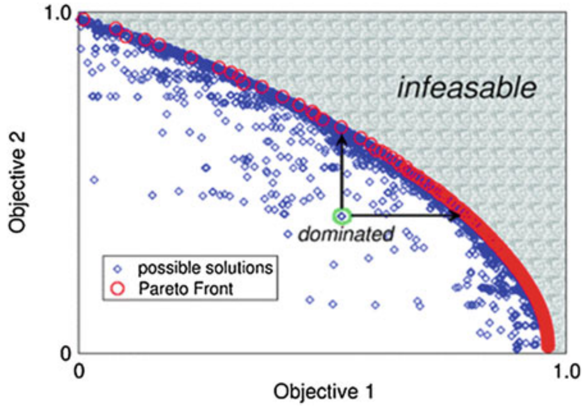


Fig. 15.6 Pareto front with two objective functions

Figure 15.6 shows a Pareto front defining the solutions for a two objective (F_1 and F_2) problem. Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives.

For a nontrivial multi-objective optimization problem, there does not exist a single solution that simultaneously optimizes each objective. In that case, the objective functions are said to be conflicting, and there exists a, possibly infinite number of, Pareto optimal solutions. A solution is called non-dominated, Pareto optimal, Pareto efficient or non-inferior, if none of the objective functions can be improved in value without degrading some of the other objective values. Without additional subjective preference information, all Pareto optimal solutions are considered equally good (as vectors cannot be ordered completely). Researchers study multi-objective optimization problems from different viewpoints and, thus, there exist different solution philosophies and goals when setting and solving them. The goal may be to find a representative set of Pareto optimal solutions, and/or quantify the trade-offs in satisfying the different objectives, and/or finding a single solution that satisfies the subjective preferences of a human decision maker (DM).

15.7 MDO Architectures

It is a fact of physics that in an engineering system such as a road vehicle there are interactions among the physical phenomena and the vehicle hardware parts. These interactions make the vehicle a synergistic whole that is greater than the sum of its parts. Taking advantage of that synergy is the mark of a good design but the web of interactions is difficult to untangle. That difficulty combined with the need to partition the work into subtasks executed simultaneously to compress the project time gave rise to the practice of dividing the detailed design work into specialty areas, each area centered on a physical phenomenon, e.g. stress and strain, or on a hardware subsystem, e.g. the car suspension. The above practice has achieved its purpose of developing a broad work front and compressing project time but on the downside it impeded trade-offs across the subtasks boundaries making the design of the vehicle fall somewhat short of optimal.

The MDO has evolved as a new discipline that provides a body of methods and techniques to assist engineers in moving engineering system design closer to the global optimum. Parallel to the development of these methodologies, a number of software packages have been created to facilitate integration of codes, data, and user interface. These packages, such as FIDO, iSIGHT, LMS Optimus, and DAKOTA, are often referred to as frameworks [40].

The key concept in several of these MDO methods is a decomposition of the design task into subtasks performed independently in each of the modules, and a system-level or coordination task giving rise to a two-level optimization. In general, decomposition was motivated by the obvious need to distribute work over many people and computers to compress the task calendar time. Equally important benefit from the decomposition is granting autonomy to the groups of engineers responsible for each particular subtask in choosing their methods and tools for the subtask execution. As an additional advantage, the concurrent execution of the subtasks fits well the technology of massively concurrent processing that is now becoming available (see Chap. 4).

Several requirements exist for a framework to provide an easy-to-use and robust MDO environment:

- Provide for quick and easy linking of analysis tools. The set of analysis tools to be linked could involve such tools as COTS software (CAD, CAE, CAM), legacy (in house) codes, spreadsheets, databases, and tools to capture user's knowledge.
- Provide effective support for geographically distributed modelling and optimization, through CORBA client-server compliancy of the software tools and models, facilitating both tight and loose collaboration, ranging from OEMs, customers, suppliers and consultants.
- Access to efficient parametric study capability such as design of experiments (DOE) based procedures, including full factorial designs, fractional factorial designs (orthogonal arrays), central composite designs and Latin hypercube designs.
- Access to a full range of optimization search strategies ranging from gradient based numerical optimization, simulated annealing and GAs and most importantly, an optimization advisor that can appropriately recommend the optimization algorithm or a combination of algorithms (hybrid optimization plan) to be used for solution of the user problem.
- Access to a full range of model approximation techniques such as polynomial, Kriging, or neural networks based response surfaces, sensitivity based Taylor series linearization, and variable complexity models.

- Provide the ability to perform trade-off studies between different design responses.
- Provide support to easy description and set up of MDO problems using formal, decomposition based MDO methods such as global sensitivity equations (GSE) based Optimization, collaborative optimization (CO), and bi-level integrated system synthesis (BLISS).
- Provide the ability to account for uncertainties in design using probabilistic constraints and robust design formulations.
- Framework should provide support for parallel computing, including parallel invocations of simulation codes as well as subsystem optimizations and intelligent load balancing [41].
- Provide effective support of visualization of design data both at runtime and post-processing stages.
- Provide effective support for database management through structured query language (SQL) interface for data storage/access/manipulation both at the local (subsystem) and global (system) levels.
- The framework should be easy to use in terms of user interface for MDO, extensible for user addition of optimization solvers, scalable for large scale problem solving and provide for robust performance [42].

A brief description of some of the formal MDO architecture used to solve the system optimization problem is provided in the following sub-sections [43–58].

15.7.1 Multidisciplinary Design Feasible (MDF)

The All-in-One (A-i-O) method, also referred to as multidisciplinary feasibility (MDF), is the most common way of approaching the solution of MDO problems. In this method, the vector of DV x is provided to the coupled system of analysis disciplines and a complete multidisciplinary analysis (MDA) is performed via a fixed-point iteration with that value of x to obtain the system MDA output variable $y(x)$ that is then used in evaluating the objective $f(x, y(x))$ and the constraints $c(x, y(x))$. The optimization problem is:

$$\min f(z, x, y(x, y, z)) \tag{15.15}$$

With respect to: z, x and subject to: $c(z, x, y(x, y, z)) \geq 0$

If a gradient-based method is used to solve the above problem, then a complete MDA is necessary not just at each iteration but at every point where the derivatives are to be evaluated. Thus, attaining multidisciplinary compatibility can be prohibitively expensive in realistic application. Figure 15.7 shows the data flow in an A-i-O analysis and optimization. The different disciplines are considered as a single monolithic analysis. This is conceptually very simple, and once all disciplines are coupled to form one single MDA module, one can use the same techniques that are

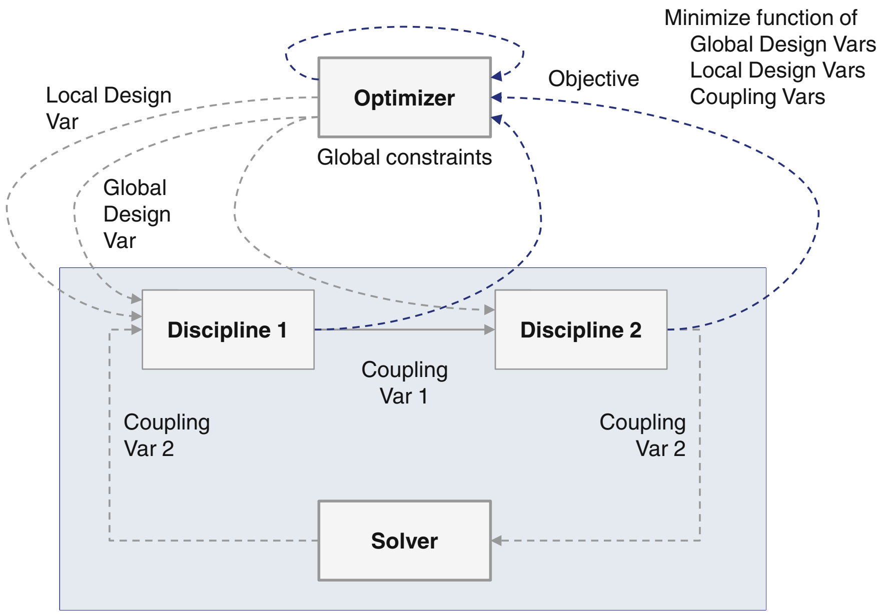


Fig. 15.7 MDF architecture

used in single discipline optimization. One of the disadvantages of this approach is that the solution of the one system might be very costly and does not exploit the potentially weak coupling between some of the disciplines that would enable the division into different analyses modules that might run in parallel. The only opportunity for parallelizing the optimization procedure would be the use of different processes for each member of the population when using a GA or running the analyses for different design points when calculating gradients by finite differencing or when evaluating the points for a response surface.

15.7.2 Individual Discipline Feasible (IDF)

The IDF formulation provides a way to avoid a complete MDA at optimization. IDF maintains individual discipline feasibility, while allowing the optimizer to drive the individual disciplines to MDF and optimality by controlling the interdisciplinary coupling variables. In IDF, the specific analysis variables that represent communication, or coupling, between analysis disciplines are treated as optimization variables and are in fact indistinguishable from DV from the point of view of a single analysis discipline solver. The IDF architecture is shown in Fig. 15.8.

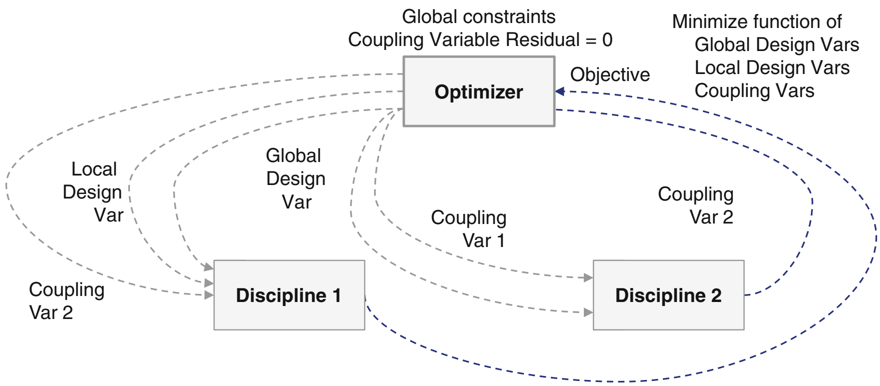


Fig. 15.8 IDF architecture

15.7.3 Simultaneous Analysis and Design (SAND)

This approach optimizes the design and solves the governing equations at the same time by posing the problem as:

$$\min f(z, x, y) \tag{15.16}$$

With respect to: x, y, z, subject to: $c(z, x, y(x, y, z)) \geq 0, \quad R(x, y, z) = 0$

SAND is not inherently multidisciplinary and can also be used for single discipline optimization problems. It can be very efficient since we solve the whole problem at once, but if a very efficient analysis is already in place, it is usually not worthwhile to use SAND. To implement SAND, one needs to calculate the residual of each governing equation.

15.7.4 Optimizer-Based Decomposition (OBD)

The main idea of this method is to use the optimizer to enforce inter-disciplinary compatibility. Instead of iterating the MDA to converge the coupling variables y, these coupling variables are given by the optimizer as a guess, or target, y^t. The new optimization problem can be written as:

$$\min f(z, x, y(x, y^t, z)) \tag{15.17}$$

With respect to: x, y^t, z, subject to: $c(z, x, y(x, y^t, z)) \leq 0, \quad y_i^t - y_i(x, y^t, z) = 0$

The number of DV has increased, and is equal to the number of original DV plus the number of coupling variables. This increases the size of the optimization problem, but conveniently decouples all the analyses, which can now be solved in parallel without intercommunication. Note that when using gradient-based optimization, the gradients $\partial f/\partial y^t$ and $\partial c/\partial y^t$ must also be calculated.

15.7.5 Collaborative Optimization (CO)

The CO architecture, shown in Fig. 15.9, is designed to promote disciplinary autonomy while achieving interdisciplinary compatibility. The optimization problem is decomposed into optimization subproblems corresponding to the different disciplines. Each subproblem is given control over its own set of local DV, is responsible for satisfying its own set of local constraints and does not know about the other disciplines' DV or constraints. The objective of each sub-problem is to agree on the values of the coupling variables with the other disciplines. A system-level optimizer is used to coordinate this process while minimizing the overall objective. The system level optimization problem can be stated as:

$$\min f(z^t, y^t) \tag{15.18}$$

With respect to: z^t, y^t, subject to: $j_i^*(z_i^t, z_i^*, y^t, y_i^*(x_i^*, y^t, z_i^*)) = 0, \quad i = 1, \ldots, N$ where N is the number of disciplines, and the subscript * represents the results from the solution of the i^h discipline optimization sub-problem:

$$\min j_i(z_i^t, z_i, y^t, y(x_i, y^t, z_i)) = \Sigma\left(1 - \frac{z_i}{z_i^t}\right)^2 + \Sigma\left(1 - \frac{y_i}{y_i^t}\right)^2 \tag{15.19}$$

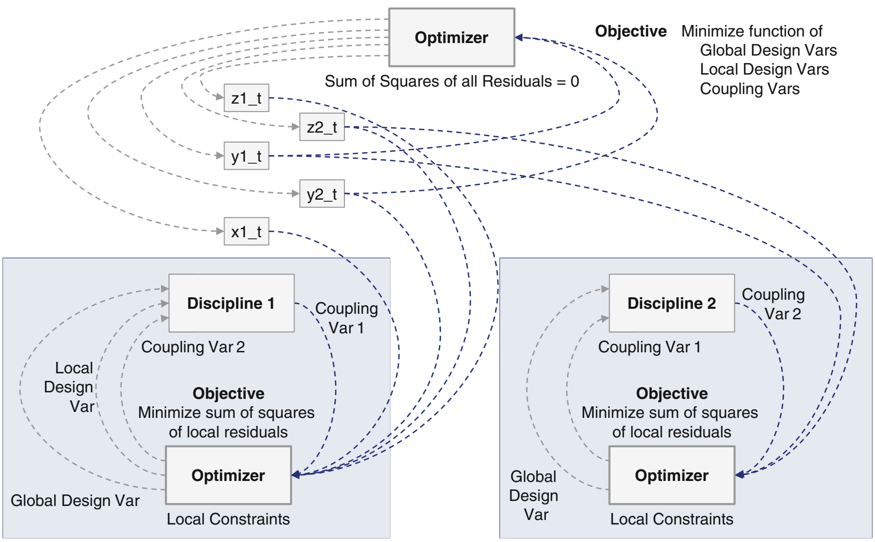


Fig. 15.9 Collaborative optimization architecture

With respect to: z_i, x_i, subject to: $c_i(x_i, z_i, y_i(x_i, y^t, z_i)) \geq 0$, where c is the vector of constraints for the i^h discipline. J is a measure of interdisciplinary discrepancy that we want to drive to zero at the system level. The solution of this sub-problem returns j_i^*. Note that post-optimality sensitivities are needed.

15.7.6 Concurrent Subspace Optimization (CSSO)

The CSSO method is also a decomposition-based strategy that allows for the disciplines to run decoupled from each other. Again, the multiple subspace optimization problems are driven by a system-level optimizer that provides overall coordination. Each sub-problem in CSSO uses approximations to non-local disciplinary coupling variables to estimate the influence of these variables on the system-level objective and constraints. The subspace optimization problem for the i^h discipline is given by:

$$\min f(z, x, \tilde{y}_j(z_i, x_i), y_i(x_i, \tilde{y}_j, z)) \tag{15.20}$$

With respect to: z_i, x_i, subject to: $c(x_i, z, \tilde{y}_j, (z_i, x_i), y_i(z_i, x_i, \tilde{y}_j)) \leq 0$, where $j \neq i$ and $y_i = (z, x_j)$ are the approximations to the other disciplines' coupling variables, or states. These approximations can be made using response surfaces. The system-level optimizer solves the following problem:

$$\min f(z, x, \tilde{y}(z, x)) \tag{15.21}$$

With respect to: z, x, subject to: $c(z, c, \tilde{y}(z, x)) \leq 0$.

After each iteration of the system-level optimizer, a MDA is performed to update the model which gives the approximate response of all coupling variables $\tilde{y}$.

15.7.7 Bi-Level Integrated System Synthesis (BLISS)

The recently introduced BLISS method uses a gradient-guided path to reach the improved system design, alternating between the set of modular design subspaces (disciplinary problems) and the system level design space. BLISS is an A-i-O like method in that a complete system analysis performed to maintain MDF at the beginning of each cycle of the path. With BLISS, the general system optimization problem is decomposed into a set of local optimizations dealing with a large number of detailed local DV (X) and a system level optimization dealing with a relatively small number of global variables (Z) in comparison with the other MDO methods. In optimization it is useful to distinguish between X and Z because:

- The X variables are associated with individual components and, therefore, they tend to be clustered. Also, the constraints they govern directly, e.g. the stringer buckling in built-up, thin-walled structures typical of aerospace vehicles, tend to be highly nonlinear. The total number of the X variables in a typical airframe is in thousands but their number in an individual substructure is likely to be quite small.
- The number of Z variables is much smaller than the total number of X variables.
- Nonlinearity of the overall behavior constraints, such as displacements, with respect to X and Z tends to be weaker than that of the local strength and buckling constraints.

With BLISS, the solution of the system level problem is obtained using either (i) the optimum sensitivity derivatives of the behavior/state (Y) variables with respect to system level DV (Z) and the Lagrange multipliers of the constraints obtained at the solution of the disciplinary optimizations, or (ii) a response surface constructed using either the system analysis solutions or the subsystem optimum solutions.

15.8 Case Studies in Multidisciplinary Design Optimization

15.8.1 Optimization of Automotive Structures Under Multiple Crash and Vibration Design Criteria

This design problem is aimed at reducing the overall mass of a vehicle by focusing on a group of structural components that are influential in both energy absorption (crashworthiness) and vehicle stiffness (vibration) [9]. Through a preliminary analysis, 22 components were selected as highlighted in Fig. 15.10. These components have a combined mass of 105.25 kg for 8 % of the crash-model mass at 1,333 kg and approximately 45 % of the vibration-model mass at 233 kg. Due to the vehicle model symmetry, the 22 components are represented by 15 wall-thickness DV denoted by $x1$ through $x15$. The 22 components contribute to 42, 27 and 36 % of the total energy absorbed in full frontal impact (FFI), offset frontal impact (OFI) and side impact (SI), respectively. In this study, the scope to sizing optimization focused on a subset of components that show considerable influence on both crash and vibration characteristics of the vehicle. The design optimization problem is formulated as:

$$\min f(x) \tag{15.22}$$

Subject to: $g_i(x) = R_i(x) - R_i^b(x) \leq 0$ with $i = 1, \ldots, 8$, $g_i(x) = R_i^b(x) - R^b(x) \leq 0$, with $i = 9, \ldots, 14$, $0.5x_j^b \leq x_j \leq 1.5x_j^b$ with $j = 1, \ldots, 15$

Where the objective function $f(x)$ represents the total mass of the selected components shown in Fig. 15.13. In the first group of design constraints, R_i, $i = 1, 8$ represent Toeboard Intrusion, Dash Intrusion for FFI and OFI, Door Intrusion for SI

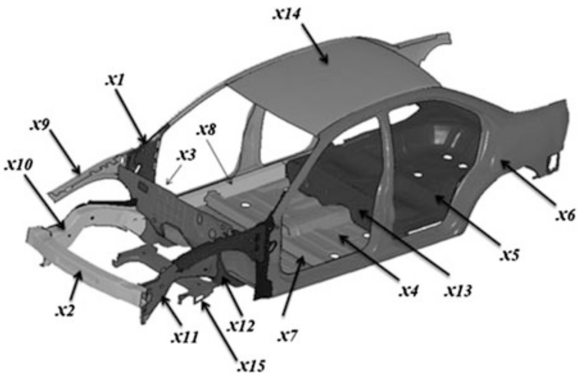


Fig. 15.10 Selected vehicle components and associated design variables [9]

in all three scenarios; all of these responses are required to be no greater than the corresponding values in the baseline model denoted by R_i^b, i = 1, 8. In the second group, R_i, i = 9, 11 represent the internal energy absorbed by the 22 components combined in the three crash scenarios whereas R_i, i = 12, 14 represent the three selected natural frequencies of the vibration model, with all required to be no less than the corresponding values in the baseline model. The design space is defined by 15 DV that represent the wall thicknesses of the components, with each bounded to within ±50 % of the respective baseline value. With the response surrogate models developed, the optimization problem was solved using SQP. Given the gradient-based search approach in SQP and the non-convex nature of the combined crash–vibration vehicle optimization problem, the problem was solved using 15 randomly selected initial design points with the best result corresponding to the optimum design defined in Table 15.1.

The objective function history showed 16 iterations for finding the optimum design point. A complete iteration refers to solution of the direction finding QP and step size associated with SQP. The optimization took a total of 163 analysis calls and approximately 20 min for the process to complete. The optimum mass was 101.49 kg for the 22 selected components in comparison to the baseline mass of 105.25 kg for a reduction of approximately 3.6 %.

Table 15.2 shows that the optimum design based on crashworthiness requirements alone reduces the overall vehicle stiffness as indicated by the frequency reduction of 6.4 % in the first mode, 5.7 % in second mode and 3.9 % in third mode. Frequencies of the current optimised design are the same as those in the baseline design. Out of 15 DV in the crash–vibration vehicle optimum, nine have increased and six have decreased relative to the respective baseline values with design variable five reaching its lower bound.

The general assessment of the results found in this study is that the crash and vibration responses are in competition. Vehicle components have to change thickness in such a way that both criteria are satisfied while weight is minimised. This is evident by the significant difference in optimised mass of the designs using

Table 15.1 Design variable bounds and optimum values [9]

Component	Lower bound (mm)	Baseline (mm)	Upper bound (mm)	Optimum (mm)
1 A-Pillar	0.806	1.611	2.417	1.471
2 Front bumper	0.978	1.956	2.934	2.169
3 Firewall	0.368	0.735	1.103	0.913
4 Front floor panel	0.353	0.705	1.058	0.560
5 Rear cabin floor	0.353	0.706	1.059	0.387
6 Outer cabin	0.415	0.829	1.244	0.897
7 Seat reinforcement	0.341	0.682	1.023	1.009
8 Cabin mid-rail	0.525	1.050	1.575	1.287
9 Shotgun	0.762	1.524	2.286	1.670
10 Inner side rail	0.948	1.895	2.843	1.694
11 Outer side rail	0.761	1.522	2.283	1.654
12 Side rail extension	0.948	1.895	2.843	1.952
13 Rear plate	0.355	0.710	1.065	0.668
14 Roof	0.351	0.702	1.053	0.815
15 Suspension frame	1.303	2.606	3.909	1.923

Table 15.2 Comparison of the baseline and optimum model [9]

Response	Baseline	Optimum	Diff (%)
FFI toe int (mm)	157.07	160.29	2.05
FFI dash int (mm)	122.06	118.30	−3.08
FFI accel (g)	63.51	59.12	−6.91
FFI int eng (kJ)	62.31	62.35	0.06
SI door int (mm)	313.93	311.09	−0.90
SI accel (g)	47.88	47.71	−0.36
SI Int eng (kJ)	22.37	23.51	5.10
OFI toe int (mm)	273.48	229.29	−16.16
OFI dash int (mm)	246.94	200.52	−18.80
OFI accel (g)	35.02	33.91	−3.17
OFI int eng (kJ)	39.42	41.46	5.17
Frq1 (Hz)	35.39	35.39	0.00
Frq2 (Hz)	36.23	36.23	0.00
Frq3 (Hz)	38.37	38.37	0.00
Mass (kg)	105.25	101.49	−3.60

crashworthiness and vibration, 101.49 kg, and crashworthiness alone, 88 kg. Adding vibration considerations to the optimization problem produced a design with less weight reduction but without sacrificing structural rigidity.

15.8.2 *Multidisciplinary Design Optimization of a Regional Aircraft Wing Box*

The structural design of an airframe is determined by multidisciplinary criteria (stress, fatigue, buckling, control surface effectiveness, flutter and weight etc.) [59]. Several thousands of structural sizes of stringers, panels, ribs etc. have to be determined considering hundreds of thousands of requirements to find an optimum solution, i.e. a design fulfilling all requirements with a minimum weight or minimum cost respectively. MDO techniques were successfully applied in sizing the wing boxes of the newly developed regional jet family. Figure 15.11 shows how the MDO process has been organized based on MSC Nastran SOL 200. Before the numerical optimization loop can be started, the design must be parameterized and all disciplines must make available their analysis models and design criteria. The wing box sizes can be parameterized by simply assigning DV to the FE-properties (cross-sections, thicknesses). The linking scheme between FE-properties and the independent DV is represented by the Design Model and it is based on constructive, manufacturing as well as numerical considerations.

Structural Analysis provides all relevant structural responses based on the analysis models and the current set of DV. The Sensitivity Analysis calculates the first derivatives of all responses with respect to the independent DV. A very important feature of MSC NASTRAN is the External Server, which allows the integration of user-defined design criteria described by Fortran routines. It therefore can be used to integrate various detailed design constraints, which are dependent on NASTRAN responses (stresses, displacements etc.). All detailed wing buckling

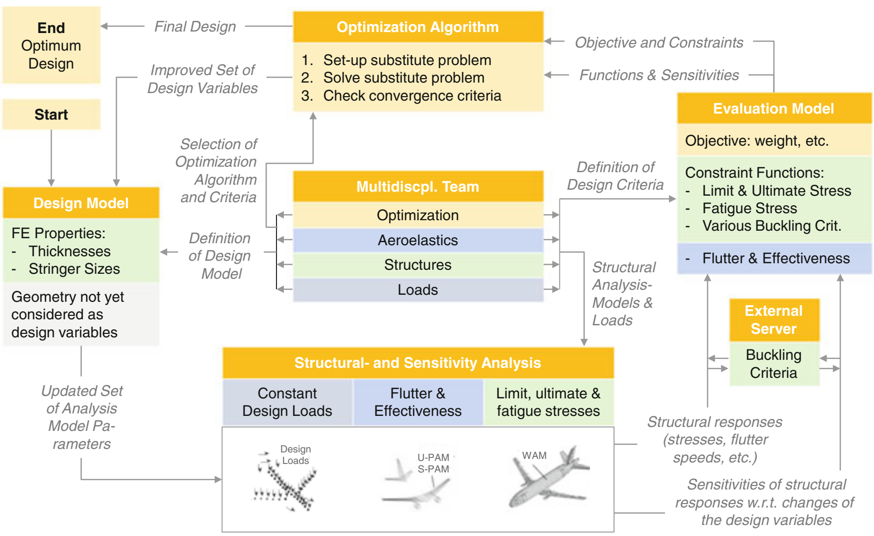


Fig. 15.11 Wing structural design process with multidisciplinary design optimization [59]

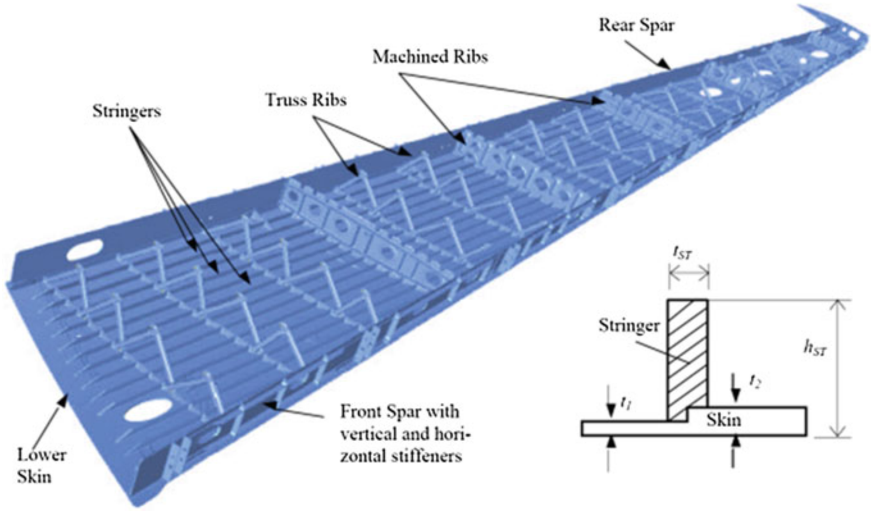


Fig. 15.12 General layout of the outer wing box [59]

criteria (skin, stringer, and column buckling and stringer crippling) have been implemented within this External Server. The objective function and all constraints are mathematically defined in the Evaluation Model based on structural responses. They are then transferred to the optimization algorithm to find an improved set of DV. This set is converted into a new set of FE-Properties in order to initiate the next cycle. As a result of the non-linear relationship between the constraints and DV, the full process must be repeated several times until an optimum design is found.

Figure 15.12 shows the lower panel, the spars and the internal ribs of the outer wing box. The panels consist of a skin stiffened by rectangular stringers. The number of stringers decreases from inboard to outboard due to wing taper. Ribs are connected both to spars and panels. The panels and spars carry global bending and torsional loads, whilst the primary function of ribs is to stabilize the whole structure and transfer the local air load into the wing box. Since the panels and the spars are machined from solids, the sizes of skin and stringers can change between each pocket surrounded by two stringers and two ribs. It is even possible to have a varying skin thickness or varying stringer height within a pocket to provide the locally required strength and stiffness with a minimum weight. This results in several thousands of independent parameters defining the whole wing box design.

The level of meshing detail of the wing model is shown in Fig. 15.13. This model is the same finite element model that is typically used for sizing by traditional methods. The wing box model mainly consists of Shell and Beam elements representing skin and stringers/stiffeners, respectively. Combining the wing box with fuselage and empennage FE models results in a Whole Aircraft Shell FE-Model (WAM) of approximately 250,000 degrees of freedom.

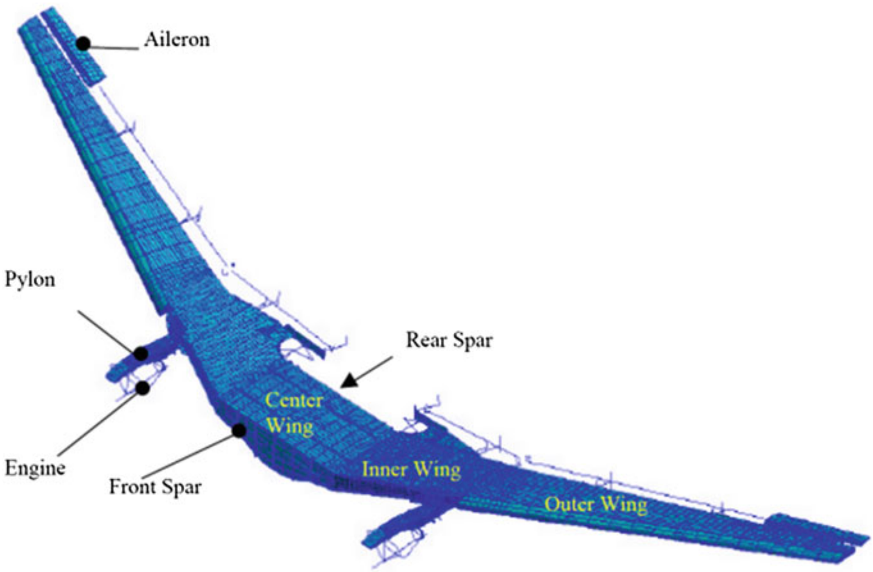


Fig. 15.13 FE-Model of the wing (93,000 DOF) [59]

The most important structural sizes of the wing box comprise the skin thickness and the stringer height and thickness. This applies to the panels as well as to the spars. Linear equations define the relationship between the independent DV and the FE-Properties representing skin and stringers sizes. For the purpose of applying buckling constraints, the upper and lower surfaces of the wing are subdivided into so called Buckling Fields. Each buckling field consists of the finite element mesh between two adjacent span wise ribs and two chord wise adjacent sets of stringers. Mechanically speaking, this corresponds to each stiffened sub-panel on the wing. The skin elements within each buckling field were linked together and represented by a single design variable.

The same applies to the stringer properties. The stringer offset and the second moment of inertia are updated after the optimization before the analysis of the new sizes takes place. The overall design model of the whole wing was structured corresponding to the major wing sections. Each of these components was subdivided again into upper and lower panels, front and rear spar, as well as skin and stringers. With this arrangement the total number of DV reached 2,515. Minimum and maximum sizes due to manufacturing or lightning protection were considered as lower and upper bounds for the FE-Properties. Special PATRAN command language (PCL) tools were developed to automate the creation and update of all corresponding design model input data for Nastran SOL 200.

The mathematical objective of the optimization process is to find a minimum feasible weight. All relevant wing box sizing criteria comprising of limit, ultimate and fatigue stresses, buckling criteria, manufacturing requirements, control surface

Table 15.3 Wing box design constraints [59]

Structure	Constraint type	Center	Inner	Outer	Load cases	Constraints
Skin elements	von-Mises stress	416	1,132	562	96 ultimate	202,560
Stringer and horizontal stiffener elements	Axial, tension and compression stress	476	985	622	96 ultimate	199,488
Spar web elements	Shear stress	148	525	280	96 ultimate	91,488
Buckling field skin	Panel buckling	147	251	364	96 ultimate	75,552
Buckling field skin	Crippling	147	251	364	96 ultimate	75,552
BF stringers	Stringer buckling	147	251	364	96 ultimate	75,552
BF skin and stringer	Euler buckling	147	251	364	96 ultimate	75,552
Lower panel skin	Principle stress	384	1042	508	3 fatigue	5,502
Panel joints	Principle stress	20	108	42	3 fatigue	510
Spar web elements	Principle stress		408		3 fatigue	1,224
Height of adjacent stringers	Maximum step size	120	199	115		434
Stringer thickness to height ration	Minimum ration	431	995	538		1,964
Outer wing box skin	Aileron effectiveness	3 times cases (zero aileron effectiveness)				3
Inner wing box skin	Lowest flutter speed	1 flutter speed limit				1
Total number of constraints						805,402

effectiveness and flutter criteria were applied in the form of in-equality constraints. The buckling constraints were communicated to NASTRAN during the optimization process by the External Server. Fatigue stress constraints were applied to all fatigue sensitive areas of the wing box. These areas included the lower skin panels, major wing box joints (inner and outer wing joint, lower front and rear panel joints), front spar web at the pylon attachment and rear spar web at the landing gear attachment. Due to manufacturing requirements, a minimum stringer thickness to height ratio had to be adhered to. Furthermore, the relative step size of the stringer height was limited in spanwise direction to prevent excessive out-of-plane bending stresses. Table 15.3 gives an overview of all constraints.

The aileron effectiveness constraint is incorporated via a roll performance criterion which is required to be greater than or equal to zero at maximum True Air Speed. A set of three trim cases, i.e. pairs of Mach number and dynamic pressure, were defined from which, on an empirical basis, the zero effectiveness curve can be extrapolated to maximum true air speed by a 2nd order polynomial.

The flutter constraint is defined such that the lowest flutter speed, i.e. a flutter mode with zero damping, must not be lower than a prescribed limit velocity which depends on the flight altitude. All normal modes up to 50 Hz are taken into account in the flutter analysis using the PK-method. The range of air speeds used for the flutter response is limited to a minimum required set. Because of the high computational effort required for flutter optimization, a pre-selection of very few critical flutter cases is indispensable. In order to get an indication for these cases, a

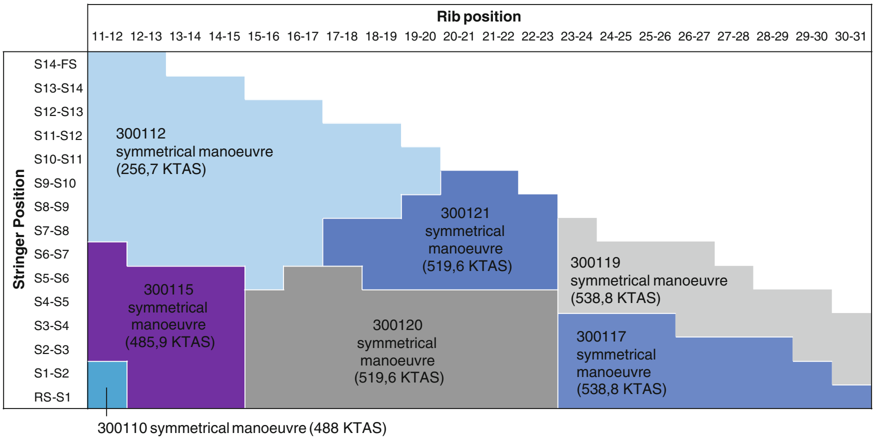


Fig. 15.14 Critical load cases, outer wing upper panels, column buckling criteria [59]

comprehensive flutter check covering the entire flight regime (i.e. a systematic variation of payload mass, fuel mass and flight level) is performed preceding the optimization runs.

A valuable means of displaying the results is shown in Fig. 15.14. In this figure, the driving load cases that design a given section with respect to column buckling of the outer wing are displayed. The driving cases are resulting from symmetrical maneuvers at different speeds, altitudes, flap settings etc. Similar plots for other wing sections and other buckling criteria are also produced. In order to satisfy the aileron reversal constraint the stiffness of the outer wing was locally increased. The skin thicknesses obtained from static optimization were taken as lower bounds. Significant changes are essentially restricted to a zone reaching diagonally from the aileron attachment area inboard to the leading edge, close to the inner wing connection. Similar results were obtained for the lower skin.

15.9 Discussion and Conclusions

MDO is at a crossroad. The focus of MDO has shifted dramatically over the past 25 years as researchers are finding new ways to use MDO methods and tools on a wide array of problems. The potential of MDO has been illustrated in this chapter with a few case studies. A strong research focus in MDO remains to resolve a number of issues that remain an impediment in implementing MDO in all levels of design an development. The major challenges in MDO integration are [60–64]:

- Integrating the designers' skills and experience in the design process. This makes the optimization task difficult to model in an algorithmic form.
- Companies have their own legacy and embedded design improvement processes and tend to resist the implementation of new optimization systems.

- Acquisition and maintenance of hardware and software can be costly.
- Handling large scale qualitative design spaces. It would be ideal to handle quantitative and qualitative information together within one framework.
- Interfaces between feature-based parametric CAD models and optimization models with automatic bi-directional conversions do not exist at present.
- Recently there is interest in design optimization within a dynamic environment. Research is required to extend this to multi-objective design optimization.
- Stochastic optimization, like GAs, is contradictory to conventional deterministic thinking, so how can the user select the most effective technique?
- Scalability is a major challenge for complex systems design optimization. Large-scale design optimization must deal with the complexity.
- There is a lack of understanding about the interaction between components and their behaviors. This may lead to results that cannot be explained.
- Uncertainty is another major challenge for complex systems design optimization. Robust design optimizations are addressing this issue.

There are three major areas of improvement when it comes to use of computing to address engineering design optimization: improve efficiency and speed of optimization and effective use of human knowledge. Large-scale optimization will require more research in topology design, computational power and efficient optimization algorithms. Emergent computing techniques such as grid computing, swarm intelligence and quantum computing improve efficiency and speed of the optimization. Future success of MDO is in application of expert knowledge with existing and emergent algorithmic and computing approaches to large-scale designs, supported by education on optimization.

References

1. de Weck O, Willcox K (2010) Multidisciplinary system design optimization. MIT ESD.77/16.888
2. Bloebaum CL (2013) The role of MDO in the design for complex engineered systems (DCES). In: 9th MDO specialist conference, 11 Apr 2013
3. Schuhmacher G (2008) Numerical optimization methods in the aerospace design process—civil and military applications and benefits. In: 2nd European hyperWorks technology conference, Strasbourg, France, 30 Sept–1 Oct 2008
4. Guo J, Guadagni L (2012) Multidisciplinary design optimization for concurrent engineering of space systems. In: SECESA2012, Lisbon, 17–19 Oct 2012
5. Neufeld D, Chung J, Kamaran B (2008) Development of a flexible MDO architecture for aircraft conceptual design. In: EngOpt 2008—international conference on engineering optimization, Rio de Janeiro, Brazil, 1–5 June 2008
6. Campana EF et al (2009) New global optimization methods for ship design problems. Optim Eng (2009) 10:533–555. doi:10.1007/s11081-009-9085-3
7. Di Pasquale E, Gielczynski G (2010) Multi-disciplinary optimization of railways systems. In: International conference on integrated design and manufacturing in mechanical engineering, Bordeaux, France, 20–22 Oct 2010

8. Breitkopf P, Filomeno Coelho R (eds) (2010) Multidisciplinary design optimization in computational Mechanics. Wiley, New York, ISBN 978-1-84821-138-4
9. Kiani M et al (2013) Surrogate-based optimization of automotive structures under multiple crash and vibration design criteria. Int J Crashworthiness 18(5):473–482
10. Bérend N, Bertrand S (2009) MDO Approach for early design of aerobraking orbital transfer vehicles. Acta Astronaut 65:1668–1678
11. Schoofs AJG (1993) Structural optimization history and state-of-the-art, topics in applied mechanics. Kluwer Academic Publishers, Dordrecht, pp 339–345
12. Schmit LA (1960) Structural design by systematic synthesis. In: Proceedings of the 2nd conference on electronic computation, ASCE, New York, pp 105–132
13. Venkayya VB (1978) Structural optimization: a review and some recommendations. Int J Numer Meth Eng 13:203–228. doi:10.1002/nme.1620130202
14. Serna1 A, Bucher C (2009) Advanced surrogate models for multidisciplinary design optimization. In: 8th Weimar Optimization and Stochastic Days
15. Boussouf L (2011) Surrogate based optimization for multidisciplinary design. In: SAE aerospace technology conference, Toulouse, France, 18–21 Oct 2011
16. Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer Science + Business Media, New York, ISBN-13: 978-0387-30303-1
17. Venter G (2010) Review of optimization techniques, encyclopedia of aerospace engineering. Wiley Ltd, Hoboken
18. Roy R, Hinduja SH, Teti R (2008) Recent advances in engineering design optimization: challenges and future trends. CIRP Ann Manufact Technol 57(2008):697–715
19. Waziruddin S, Brogan DC, Reynolds Jr PF (2004) Coercion through optimization: a classification of optimization techniques. In: 2004 fall simulation interoperability workshop, Orlando
20. Moldoveanu G, Abaluta O (2009) Multidisciplinary optimization in urban services management. Theor Empirical Res Urban Manage 1(10)
21. Lee HJ, Lee JW, Lee JO (2009) Development of web services-based multidisciplinary design optimization framework. Adv Eng Softw 40:176–183
22. Schittkowski K, Yuan Y (2007) Sequential quadratic programming methods. Wiley Encyclopedia of Operations Research and Management Science. doi:10.1002/9780470400531.eorms0984
23. Boggs PT, Tolle JW (1995) Sequential quadratic programming. Acta Numer 4:1–51
24. Lasdon LS, Fox RL, Ratner MW (1973) Nonlinear optimization using the generalized reduced gradient method. In: Office of naval research, technical memorandum no. 32
25. Michalewicz Z, Janikow CZ (1991) Genetic algorithms for numerical optimization. Stat Comput 1(2):75–91
26. Fernandes de Oliveira R et al (2008) Genetic optimization applied in conceptual and preliminary aircraft design. In: XVII Congresso e Exposição Internacionais da Tecnologia da Mobilidade, São Paulo, Brasil, 7–9 Oct 2008
27. Hart CG, Vlahopoulos N (2009) Integrating a particle swarm optimizer in a multi-discipline design optimization environment for conceptual ship design. In: SAE world congress and exhibition, Detroit, MI, 20–23 Apr 2009
28. Schutte JF et al (2004) Parallel global optimization with the particle swarm algorithm. Int J Numer Meth Eng 61(13):2183–2387
29. Peri D, Fasanoy G, Dessi D, Campana EF (2008) Global optimization algorithms in multidisciplinary design optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, Victoria, Canada, 10–12 Sept 2008
30. Strobbe T, Pauwels P, Verstraeten R, De Meyer R (2011) Metaheuristics in architecture. Int J Sustain Constr Des 2(2):190, ISSN 2032-7471
31. Gonzalez LF, Periaux J, Srinivas K, Whitney EJ (2004) Evolutionary optimization tools for multi-objective design in aerospace engineering: from theory to MDO applications. In: Evolutionary algorithms and intelligent tools in engineering optimization, CIMNE, Barcelona

32. Depince P, Guedas B, Picard J (2007) Multidisciplinary and multiobjective optimization: comparison of several methods. In: 7th world congress on structural and multidisciplinary optimization, Seoul, 21–25 May 2007
33. Kesseler E, van Kan WJ (2006) Multidisciplinary design analysis and multi-objective optimization applied to aircraft wing. In: National aerospace laboratory NLR, NLR-TP-2006-748
34. Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. J Struct Multi Optim 26:369–395
35. Li M (2007) Robust optimization and sensitivity analysis with multi-objective genetic algorithms: single- and multi-disciplinary applications. PhD thesis, University of Maryland
36. Zitzler E, Laumanns M, Bleuler S (2004) A tutorial on evolutionary multiobjective optimization. In: Metaheuristics for multiobjective optimization, vol 535. Springer, Berlin. doi:0.1007/978-3-642-17144-4_1
37. Yang F, Bouchlaghem D (2010) Genetic algorithm-based multiobjective optimization for building design. Archit Eng Des Manage 6:68–82. doi:10.3763/aedm.2008.0077
38. Fantini P (2007) Effective multiobjective MDO for conceptual aircraft design—an aircraft design perspective. PhD thesis, Cranfield University
39. Pierret S (2005) Multi-objective and multi-disciplinary optimization of three-dimensional turbomachinery blades. In: 6th world congresses of structural and multidisciplinary optimization, Rio de Janeiro, Brazil, 30 May–3 June 2005
40. Salas AO, Townsend JC (1998) Framework requirements for MDO application development. In: 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, St. Louis, MO, 2–4 Sept 1998
41. Manolache F, Costiner S (2002) Parallel processing approaches for multi-disciplinary optimization algorithms. Carnegie Mellon University, Department of Mathematical Sciences, Center for Nonlinear Analysis
42. Gould N, Orban D, Toint P (2005) Numerical methods for large-scale nonlinear optimization. Acta Numer 14:299–361
43. Martins JRRA, Lambe AB (2013) Multidisciplinary design optimization: a survey of architectures. AIAA J 51(9):2049–2075
44. Cramer EJ et al (1993) Problem formulation for multidisciplinary optimization. In: AIAA symposium of multidisciplanary design optimization
45. Mainini L, Maggiore P (2012) Multidisciplinary integrated framework for the optimal design of a jet aircraft wing. Int J Aerosp Eng 2012(750642):9, doi:10.1155/2012/750642
46. Ajmera HC, Mujumdar PM, Sudhakar K (2004) MDO architectures for coupled aerodynamic and structural optimization of a flexible wing. In: 45th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, Palm Springs, CA, 19–22 Apr 2004
47. Sobieszczanski-Sobieski J, Barthelemy JF (1985) Improving Engineering system design by formal decomposition, sensitivity analysis, and optimization. In: NASA technical memorandum 86377
48. Kodiyalam S, Sobieszczanski-Sobieski J (2001) multidisciplinary design optimization—some formal methods, framework requirements, and application to vehicle design. Int J Veh Des 25 (1/2):3–22
49. Sobieszczanski-Sobieski J (1990) Sensitivity analysis and multidisciplinary optimization for aircraft design: recent advances and results. AIAA J Aircr 27(12):993–1001
50. Kroo I, Manning V (2000) Collaborative optimization: status and directions. In: 8th AIAA/NASA/ISSMO symposium on multidisciplinary analysis and optimization, Long Beach, CA, 6–8 Sept 2000
51. Braun RD, Kroo IM (1995) Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In: Proceedings of the ICASE/NASA langley workshop on multidisciplinary design optimization, Hampton, VA, 13–16 Mar 1995
52. Du X, Chen W (2001) A hierarchical approach to collaborative multidisciplinary robust design, Department of Mechanical Engineering, University of Illinois at Chicago, Chicago

53. Braun RD, Gage P, Kroo I, Sobieski I (1996) Implementation and performance issues of collaborative optimization. NASA/TM-2004-213192
54. Simpson TW, Martins JRRA (2011) Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop. J Mech Des 133 (10):101002. doi:10.1115/1.4004465
55. Ryan A (2007) A Multidisciplinary approach to complex systems design. In: PhD thesis, University of Adelaide
56. Sobieski J (2010) A Perspective on the state of multidisciplinary design optimization (MDO). In: Workshop on MDO, Fort Worth, 16 Sept 2010
57. Sobieszczanski-Sobieski J, Haftka RT (1995) Multidisciplinary aerospace design optimization: survey of recent developments. In: 34th aerospace sciences meeting and exhibition, Reno, NV, 15–18 Jan 1995
58. Peri D, Fasanoy G, Dessi D, Campana EF (2008) Global optimization algorithms in multidisciplinary design optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, Victoria, Canada, 10–12 Sept 2008
59. Schuhmacher G, Murra I, Wang L, Laxander A et al. (2002) Multidisciplinary design optimization of a regional aircraft wing box. In: 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Atlanta, GA, 4–6 Sept 2002
60. Simpson TW, Martins JRRA (2012) Advancing the design of complex engineered systems through multidisciplinary design optimization. In: NSF workshop, 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, Honolulu, Hawaii, 23–26 Apr 2012
61. Kolonay RM (2014) A physics-based distributed collaborative design process for military aerospace vehicle development and technology assessment. Int J Agile Syst Manage 7(3/4):242–260
62. van Tooren M, La Rocca G (2008) Systems engineering and multi-disciplinary design optimization. In: Curran R et al (eds) Collaborative product and service life cycle management for a sustainable world. Proceedings of the 15th ISPE international conference on concurrent engineering. Springer, London, pp 401–415
63. Sobolewski M (2014) Unifying front-end and back-end federated services for integrated product development. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 3–16
64. Ottino A, Ghodous P, Ladjal H, Shariat B, Figay N (2014) Interoperability of simulation applications for dynamic network enterprises based on cloud computing—aeronautics application. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 597–606

Chapter 16
Product Lifecycle Management

Lutz Lämmer and Mirko Theiss

Abstract Product lifecycle management (PLM) is widely understood as concept for the creation, storage, and retrieval of data, information and, ideally, knowledge throughout the lifecycle of a product from its conceptualization or inception to its disposal or recovery. PLM is seen in industry as one of the core concepts to fulfill a number of business requirements in the manufacturing industry with respect to completeness, high transparency, rapid accessibility, and high visibility of all product data during a product's lifecycle. Those requirements are related to financial aspects such as cost management and revenue growth; to the product itself like innovation, time to market, quality, and high productivity; and to regulatory aspects such as compliance and documentation. PLM is implemented by deploying IT systems such as product data management (PDM) systems and induces a high level of interoperability of related applications. With PLM, industrial companies attempt to gain advantages in shorter cycles, lower costs, better quality by avoiding errors, and misunderstanding. After reviewing basic concepts and building blocks of PLM, we provide empirical evidence of implementation scenarios and use case studies for different integrations to build up PLM solutions. We have evaluated applications in automotive, aerospace and consumer electronic industries focused on engineering design, change management, simulation data management integration and communication with partners. Emphasis is on the organizational and IT implications and the business benefit of the provided solutions.

Keywords Product lifecycle management · Product data management · Engineering release · Change management · Access management

L. Lämmer (✉) · M. Theiss
PROSTEP AG, Dolivostrasse 11, 64293 Darmstadt, Germany
e-mail: lutz.laemmer@prostep.com

M. Theiss
e-mail: mirko.theiss@prostep.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_16

16.1 Introduction

Product lifecycle management (PLM) is well established as a must in state of the art for large companies in manufacturing industries, especially in the world of automotive, aerospace, and increasingly of consumer electronics industry. This world is driven by the need for high transparency of all activities during the product creation process, clear, repeatable workflows, fast access to all product related data, distributed engineering and early supplier involvement. PLM as a concept focuses on the creation, storage, and retrieval of data, information and, ideally, knowledge throughout the lifecycle of a product from its conceptualization or inception to its disposal or recovery. Technological fundaments of PLM are product data management (PDM) systems which manage product data and knowledge during a product's lifecycle. Accompanied by powerful CAD tools, PDM builds comprehensive software systems which support all relevant roles in the product creation process, building an umbrella over all product-related activities [1].

Today's development process of attractive, complex products requires the contribution of large specialist networks (see Chap. 7). In this kind of collaboration, product data must be frequently exchanged between involved parties in digital form, with a high level of information security (see Chap. 18). Basically, PLM can also be understood as a concept for collaboration in the supply network and for managing product creation and lifecycle processes in today's networked world. Thus, PLM facilitates the acceleration of product creation process, by enabling communication on a regular basis between participants, preventing errors, and cutting the costs.

During serial production, engineering change management (ECM) is one of the key processes to be considered inside PLM. ECM is recognized as an issue that gains relatively little attention considering its importance. A particular issue is ECM within a supply chain. Suppliers are mostly loosely connected to their customers, and they are often not involved into an engineering change approval process. Many and especially late engineering changes induce high additional costs for any development project [2]. They consume one-third up to one half of the total engineering capacity and represent 20–50 % of total tool costs [3]. Thus, management of engineering change is a fundamental requirement for PLM.

After a short review on business requirements (Sect. 16.2) and related work (Sect. 16.3), we will describe the benefits of PLM (Sect. 16.4) in particular of the building blocks of PLM (Sect. 16.5). We will describe its processes and systems, its aspects, its software elements, its integrations, and its challenges. Those challenges are induced by the PLM concept due to its integrative nature. Consecutively, in Sect. 16.6, we describe empirical findings on different integration scenarios such as system integration, cross-domain integration, partner integration, and the special case of PDM system migration. Those findings are supported by the description of an integration tool in Sect. 16.7 and its application to a number of case studies in Sect. 16.8. This chapter is closed by conclusions and an outlook for further work (Sect. 16.9).

16.2 Business Requirements

PLM is seen as one of the core concepts to fulfill a number of business requirements in the manufacturing industry. Those requirements are related to financial aspects such as cost management and revenue growth; to the product itself like innovation, time-to-market, quality, and high productivity; and to regulatory aspects such as compliance and documentation. Industries demand a comprehensive concept which fulfills the following requirements [4]:

- The product is described completely and consistently in a global network composed of various systems.
- Results of real tests and experience from exploitation are part of the digital description.
- Each singular configurable product can be visualized and simulated in each version during the product lifecycle.
- The entire product creation process is accomplished in a network with suppliers and partners.
- The entire product creation process is conducted in a distributed, international environment.
- The complete information about product is available during the entire product lifecycle.

The user-friendly PDM system that realizes these requirements shall provide the following basic functionality:

- Fast, filterable access to all information which is relevant to the product creation process.
- Storage for all information which is generated by an application used in the product creation process.
- Forward of all information which is needed in the corresponding downstream processes.

Where financial aspects constitute a business case which needs to be identified for most PLM introduction projects that consume significant human effort and invested capital, PLM itself tends to increase cost transparency and therefore is an enabler of cost reduction applied to the product and its manufacturing process. Such cost reductions comprise improved communication with less effort, reliable sources of information, and standardization of processes. Spending more effort into consistent product documentation during the design phase has significant advantages in later lifecycle phases of the product such as manufacturing, change, and after sales.

PLM is implicitly a source of revenue growth which is achieved by accelerating product development and increased product variety under full cost control. PLM is seen as a preconditioning for flexible and agile development in mutually beneficial collaboration scenarios to create innovative and competitive products within shorter time, under full quality control and with a highly efficient production line. It is a competitive advantage to phase-in and phase-out partners in the supply chain and to establish working relationships fast and reliably. This is both true from a cost perspective by moving from single sourcing to competitive sourcing and from an innovation point of view by relying solely on the single source partner's capability to develop the new technology eventually or choosing partners capable of providing a sought for new technology.

As a single source of data PLM avoids erroneous manual data replication, supports traceability and manages dependencies. This opens the option for completely virtual and therefore faster and potentially cheaper product creation. Of course, this requires the virtual product creation methods to be stable, reliable and efficient. PLM integrates digital techniques and builds up a network of interconnected information. It connects data across system boundaries and is a precondition for accountability across the value chain. PLM has to become a reliable and constantly available source of information. Representing the lifecycle of a product and managing the lifecycle of its associated data is essential for cost efficient downstream processes in change management, after sales, and customer services.

PLM is a precondition for a seamless product documentation process to support all kinds of requests arising from regulatory definitions. PLM has to provide the data base for all kinds of documentation. Product documentation needs to be represented in different formats and with a well-defined status. This documentation must accompany product instances along their lifetime. This is essential for mission-critical products, like aircrafts and their engines, to keep track of the product instances themselves. Comparable requirements arise from regulations for providing spare parts or replacement instructions for the product lifetime and to provide documentation on proper end-of-life procedures.

16.3 Related Work

Eigner and Stelzer [1] define PLM solutions as "the functional and administrative backbone of IT solutions" which comprise a number of "IT systems and tools for CAD, CAM, CAE, simulation and visualization" and which were "derived in the 90s as an extension to PDM". Furthermore, they claim that PLM must be seen as a part of IT strategy to support the complete product development process and demands an integrated product data model, and technological and organizational preconditions in the enterprise.

Silcher [5] outlines differences in scoping the lifecycle by Germany-based researchers and others. Eigner [1], Schuh [6], and Vajna [7] focus, for example, on production phases but Stark [8] focuses on the product itself. He adds the integration of factory lifecycle management (FLM) and supply chain management (SCM) to the picture. The interface of the engineering and design phase with production is reviewed based on available empirical reports by Dekkers et al. [9]. He claims, for instance, the need for a more thorough understanding of the nature of collaboration through PLM between the involved disciplines and improved understanding of the networked structure of the lifecycle processes, especially the necessary feedback from production and maintenance to PLM instead of solely data consumption. In the following we will focus on the engineering, design and production planning phases of the product and the factory lifecycle and the supplier network management part of SCM.

Where frequent studies, by Abramovici [2] and follow-ups, underline the importance of PLM the actual level of PLM implementation varies. The key features of PLM systems are classified as [10]:

- backbone platform for all engineering data and engineering processes including process management and simulation tools,
- cross-domain and cross-enterprise management of systems, and
- management of virtual products and of representations of real instances during virtual try-out, for lifetime management or service purpose.

A PLM solution is based on a reference model which covers all lifecycle stages and all involved disciplines to represent a digital version of a product [11]. According to the industry wide accepted "Liebensteiner theses" (see [12]) PLM

- is a concept and not a system or a self-contained solution,
- is constituted of software modules like CAD, CAE, CAM, VR, PDM and other software tools supporting product development,
- provides interfaces to other application domains like ERP, SCM and CRM, and
- is supported by specialized service providers and software products to realize PLM concepts.

Representation of the product structure is central to PLM [13]. Consequently, PDM is seen as the core component of PLM. PLM, though, has a strong focus on products lifecycle processes. Corresponding authoring tools, workflow management capabilities and connected application areas are prerequisites of a successful realization of PLM throughout an enterprise.

16.4 Benefits of PLM

PLM becomes the challenging enabler for better, faster and innovative digitally based product development in increasingly complex enterprises. PLM manages product information from the earliest idea until the end of life of a manufactured product or a single item. Often seen as a concept to organize processes, PLM needs a strategy, dedicated organizations and well-selected software tools to allow collaborative product management along the product lifespan, across different domains to address all process requirements, and across different enterprises to allow for efficient and localized manufacturing. Business drivers for introducing advanced capabilities to manage product information are the demand for distributed development and manufacturing, service and maintenance in a globalized market both of customers and suppliers and more complex products in shorter time intervals. This is especially true for complex products like cars or airplanes but also for consumer goods which need locally and timely adaptations to regional demands.

PLM is not only an integration concept process-wise and system-wise, but mandatory for other methods and technologies such as requirements engineering, digital mock-up and variant management. By implementing a PLM solution a company is empowered to act internally as a virtual entity with an almost instant flow of information across the different departments but this methodology allows integration across enterprise borders and acting as an entity together with external partners in a so-called “Extended Enterprise” (see Chap. 7). Thus, PLM also enables supplier integration. Particular importance gets PLM in the case of a closer collaboration scheme or a company merger, when multiple internal lifecycle processes need to be integrated. With PLM the product relevant business processes gather a high level of transparency.

PLM requires the standardization of methods, interfaces and processes, and hence is a strong driver of standardization of the underlying information technology. Relevant standards in the context of PLM are hosted by international organizations like ISO, OMG, OASIS and a large number of national standard initiatives of the relevant industry associations like VDMA, VDA or VDI in Germany.

After discussing potential approaches to realize a company-wide PLM implementation we will focus on typical development challenges like choosing a suitable PLM architecture, selecting the right PLM tools, transforming processes and organizations, integrating PLM with legacy and supplier data to the expected advantages with respect to data availability, complexity management and increased flexibility in product creation and utilization of virtual techniques.

The case studies explain typical approaches of PLM in different industries like automotive, aerospace, and consumer goods to support data exchange and collaboration scenarios at different levels.

16.5 PLM Building Blocks

PLM comprises a number of concepts to manage all product related data from early idea until the end of life of the product.

16.5.1 Processes and Systems

From a process point of view PLM can be seen as a support for market analysis, product planning, product development, manufacturing, after-sales marketing, repair and services, and de-assembly or recycling. From a software point-of view it comprises originally disjoint software solutions like computer aided design, computer aided manufacturing, manufacturing engineering, project management, program management, team data management, product data management, material

Fig. 16.1 PLM domains

resource planning and enterprise resource planning domains (see Fig. 16.1). With its interfaces PLM supports, for example, change management or reporting by a seamless data flow and acts as the data backbone of a digital enterprise. The PLM vision is to grant access, wherever necessary, to a single source of data with reliable state and to represent all stages of a product lifecycle in such a way, that the static description of the product and its components are sufficient to perform all necessary management tasks without replicating data input.

16.5.2 Aspects

All different concepts or methodologies are based on a shared data model which is build up from specific product related master data management, document management and status management aspects. The shared data model is furthermore used to build-up more complex aspects like bill of material for functional, assembly, management, or sales breakdown, and like configuration management, effectivity management, change management, project and process management.

Accompanied by specific searching and grouping concepts, potentially also geometry based, efficient browsing and data manipulating processes can be supported. Those aspects can be seen as PLM specific. They are accompanied by not specifically PLM related aspects like user management, transaction management, security management, and workflow management or file management, which are more common to general software systems.

16.5.3 Software Elements

From a software architectural point of view PLM systems are usually multi-tier and candidates for SOA enabled solutions. Based on a generic data model representation in a general purpose database system multiple layers of abstract data handling and manipulating routines are assembled to fulfill predefined business methods and orchestrated by embedded workflows. A communication layer allows the seamless integration with infrastructure components like a firewall or an enterprise service bus. The representation layer may consist of a thin web layer with less functionality or of more advanced solutions based on rich client technology. It is the software vendor's choice to implement general purpose PLM functionality or just industry related business logic into the client. The first solution tends to produce rich but complicated and not easy-to-learn interfaces while the latter may have a competitive edge in specific customer niches with a neat and dedicated approach. Customization or a proprietary software development may be the solution for a carefully chosen customer specific solution.

The end-user experience is build-up from tables, forms, hierarchical lists, spreadsheets and embedded visualization of status networks, schematic figures and geometry viewers or graphical editors.

16.5.4 CAD Integrations

PLM systems do not have CAD editing facilities themselves but provide the necessary support based on associated CAD tools. This functionality is usually better supported for tools of the same provider. The reason for this is the somewhat historically motivated moving boundary between CAD and PLM. Both systems handle to certain, but not clearly separable, extent versioning, structure and configuration. Quite a few PLM related attributes are easily manageable by CAD editors. Some attributes—like calculated weight—even have their source in CAD models. The biggest challenge arises from handling references of CAD data which need to be taken into account for building corresponding PLM relationships. Such relationships need to be synchronized throughout the editing process. This task becomes even more complex for relationships between multiple CAD documents representing assemblies or derived information like drawings.

The interactions between a CAD workspace during the ongoing design process and the necessary versioning or updating methods in the PLM data store require carefully tuned integration tools which have access to internal know-how and state of both CAD and PLM data. Some PLM vendors resolve this challenge by coupling the CAD workspace to the PLM data base directly. Others just use internal application programmer's interfaces to build this bridge.

16.5.5 Challenges in PLM

Most PLM products gained an increasing complexity over the last years. This is driven by the demand to provide users more comfort and a new usage experience—especially for products of the automotive, aerospace and consumer goods industries.

Additionally, new regulations and laws drive the need to monitor environmental compatibility from early conception of a product until the recycling phase.

PLM is a methodology to manage all data during the lifecycle of a product, but PLM is not a single application. PLM is always to be seen as a network of many applications supporting different special domains during the lifecycle of a product. In result, there is a strong need to exchange and share PLM data between different applications and domains as well as within partner networks and supply chains. Consequently, these requirements lead to a number of challenges in PLM:

- Integration with legacy systems (cross-technology and cross-system)
- Integration with other applications (cross-domain)
- Collaboration with external partners (cross-company).

16.5.6 Integration

A PLM solution depends heavily on data availability and quality. Primary sources of data are authoring systems or legacy data stored in databases, spreadsheets or archives. Data need to be identified, clustered, categorized, and cleaned-up according to a chosen terminology to be revised and maintained in a PLM system. IT tools to inspect, transform and load data via well-defined interfaces will help to prevent erroneous interactive replication of data. While it is sufficient to achieve interoperability of the utilized tools, data models of the tools need to be aligned and mapped. Even if there is no need to integrate all tools into a single solution, interdependency of data models leads to a necessarily contextual integration.

If data is stored in legacy systems, the preferred solution to integrate them into a single PLM solution would be to identify use cases, map them to PLM functionality and migrate the data in a single step after applying carefully all necessary measures to ensure data quality. Often it is not possible to eliminate existing processes based on legacy systems. In such cases an attempt to restrict legacy data to read-only usage may be an option. Especially processes modifying PLM data, like versioning or editing attributes, status or associated secondary content, need to be transferred to the PLM solution. If this is not feasible, uni-directional integration is not sufficient and needs to be bi-directional to allow for an update of the legacy data from the PLM solution. Organizational measures to make the old data source obsolete, for instance by not initiating new projects in the old database, will eventually lead to a significantly longer but definitely finite life of the legacy system. Well-defined interfaces to technology and systems help to build such integrations.

16.5.7 Collaboration

The traditional way to extract a subset of data in well-known office spread-sheets and send this information to a partner is not efficient anymore. The manual effort to prepare the data before sending and to use them after sending it is too high and, of course, dramatically error-prone. Furthermore, regulatory and contractual obligations require an instantly repeatable and secure exchange process. Cross-enterprise communication is bound to even more legally binding and reliable means of communication. Coupling processes supported in different software systems demand data transformation. Securing intellectual property needs careful filtering of data to be exchanged. Round-trip scenarios ask for correlation of input and output data flows.

Today's requirements for an efficient PLM-based collaboration are the demand for a close integration of data exchange methods into the PLM systems itself. Typically, a PLM system supports data exchange with other PLM systems of the same vendor, but this needs to be adjusted to the customization of the PLM solution. Data exchange with a PLM system of a different vendor is in all cases not supported directly. Such integration comprises the following steps which differ in complexity depending on the scale of integration (Fig. 16.2):

- Collecting PLM data to be sent to a partner in the PLM system.
- Exporting selected data automatically out of the PLM system and track the transfer of the data.
- Conversion of CAD data during export to a preselected format to ensure intellectual property protection.
- Package data and sent them to a selected partner.
- Re-importing data sent back from the partner into the originally sending PLM system.

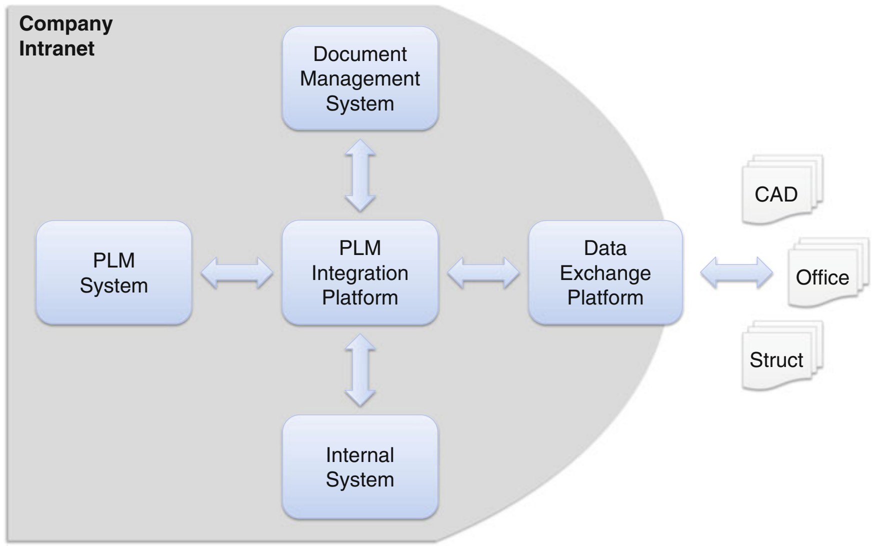


Fig. 16.2 PLM system integration and data exchange

16.6 Integration Scenarios

Already in 2005 Abramovici et al. [2] stated: "advanced PLM users will prefer integrated PLM/ERP solutions". Such integrated solutions have been detailed by Mechlinski in [14] and we will refine the business drivers and features of the identified four levels of integration below (compare overview in Table 16.1):

- TDM solution
- Integrated approach
- Best-of-breed approach
- Point-to-point integration.

Implementing commercial-off-the-shelf (COTS) PLM software as a so-called team data management system appears to be very useful if some or optimally all of the following assumptions are true:

- A single department or a small number of departments with some employees is working with product data. The data authors and the data consumers share their perspective on the product data. There are no external consumers for product related data.
- All working processes are stable and shared between the process owners. The processes are homogeneous and consistent. They are well understood and common practice. They will not change in the future.
- Accompanying processes are not based on IT support. No significant value is gained from other IT tools supporting the product development process.
- Only one single CAD system is in use. It will not be replaced in the future.
- Legacy management tools like document management or classification schemes are well understood and are easy to migrate to the new data base.
- The product management solution resides in a single network domain. There is no need for a bridge to a remote location or to external partners.

COTS solutions for integrated CAD and PDM with the right CAD tool integration of the identical software vendor are optimal in such cases. The implementation can be optimized with respect to customization efforts if it is applied out-of-the-box (OOTB). This allows a seamless migration path if new software versions evolve and additional functionality is desired. Optimal business value is offered by special pre-packaged solutions of system vendors. A *TDM solution* will come with the basic support functionality to manage CAD document based workflows for sharing design artifacts within small work groups. All users have a quite similar set of functionality available. The graphical user interface is capable of simple check-in and check-out workflows and to manage comparisons of database content within report and list views and allows spreadsheet like actions for search, replace, modify and update.

Table 16.1 Integration scenarios and their corresponding assumptions

	TDM solution	Integrated approach	Best-of-breed approach	Point-to-point integration
Involved departments	Single department within same enterprise	Cross-department solution	Cross-department solution	Cross-enterprise solution
COTS	Feasible	Feasible	N/A	N/A
CAD system	Single CAD	Single CAD	Multiple CAD	Single CAD
PDM system vendor	Identical with CAD vendor	Identical with CAD vendor	Arbitrary	Arbitrary
Partner integration	Data exchange solution based on check-in-check-out only	Portal or direct access, additional data exchange solution	Portal or direct access, additional data exchange solution	Targeted at distributed scenario with shared data
Network	Local only	Local only	Distributed locations	Targeted at distributed locations
Customization effort	Small	Moderate	Challenging	N/A

Nevertheless, the criteria mentioned above will not apply to all situations. A number of PDM integration scenarios exist to overcome problems while some or potentially all of the criteria are not met. An example for such a solution is the *integrated approach* which addresses the following assumptions:

- Many departments collaborate with identical requirements on working with product data. Data authors and data consumers share their perspective on the product data. There are some external consumers for product related data.
- All working processes are stable and shared between the process owners. The processes are homogeneous and consistent. Interfaces are well defined and understood. There exists a common practice of revising processes and the related IT tools. There will be small changes only in the future.
- Accompanying processes are based on some IT support. Their requirements will be covered by the integrated solution itself, if not, then by some customization or by data exchange via an existing interface.
- Only one single CAD system is in use with no plans to replace it in the future.
- Legacy management tools like document management or classification schemes are well understood and are easy to migrate to the new data base.
- The product management solution resides in a single network domain. The need for a bridge to a remote location or to external partners is covered by the COTS PLM solution, preferably by an integrated portal solution.

A single highly integrated PLM system—with a unique CAD system by the same PLM vendor will meet these assumptions. It depends on the ability of the enterprise to adapt to the features of the PLM system how much customization and legacy implementation, especially for interfacing with legacy, is required. More customization is associated to constantly high efforts for implementing, maintaining, testing and additionally revising the software maintenance efforts spent by the software vendor and paid for by every customer. The major advantage of such a solution is the single system appearance, which allows a high degree of standard management and training procedures out of the box as long as the customization of the software is moderate. Software updates are feasible but need special considerations for all interfaces and potentially locked-in customizations.

The PLM system needs to be highly flexible to accommodate all requirements, at least to an acceptable extent, especially for the following building blocks:

- The data model needs to be flexible and extendable. This includes, but is not restricted to, additional attributes for standard elements, new lists of values to control the population of attributes, inheritance mechanisms to derive specialized custom element types from standard elements, which need to be recognized in workflows, status and access control.
- The workflow needs to be highly flexible with respect to triggered data manipulation actions, state control and other automated processes.
- The graphical user interface must allow role-centric subsets according to specialized user profiles, like CAD engineer, PLM data manager, controller, project manager. A one-size-fits-all approach tends to be too heavyweight for the majority of the users and will reduce the number of potential or satisfied users.

Nevertheless, depending on a single vendor and his single vision on PLM, or on a monolithic software infrastructure, may have its disadvantage. If cutting edge PLM functionality and individual processes becomes a competitive asset the *best-of-breed approach* may be more appropriate. This approach addresses the following requirements:

- A large number of departments collaborate with differing requirements on working with product data. The data authors and the data consumers have different views on the product data. There are a large number of external consumers for product related data, for example in after sales and maintenance.
- The working processes are mature but not shared between the process owners. Interfaces exist, but they are not established at all levels.
- There exists a common practice of revising processes and the related IT tools. There will be constant changes in the future.
- Support processes are heavily based on IT tools. Their requirements are not covered by the PLM solution itself, but by purpose made, legacy applications which exchange data via an existing interface.
- Multiple CAD systems are in use or a change will take place in the future.
- Legacy management tools like change management, document management or classification or configuration schemes need to be supported as they are. There are no plans to integrate them into the PLM solution.

- The product management solution resides in multiple network domains to accommodate multiple sites around the world. The need for a bridge to a remote location or to external partners is covered by a specialized portal solution.

The best-of-breed approach supports individual processes optimally. Custom made solutions fit to the specific requirements of the users. Specialized CAD- and CAE-interfaces allow seamless access to required databases for the engineers and designers. Nevertheless, all solutions need to share common design models, design product structures and documents, standard parts libraries, configuration, and effectivity control. This asks for a company data model with deeply integrated interface solutions. Setting up the company-wide data model is an additional challenge. Those interfaces cause system dependencies which need to be managed by a common versioning process.

To avoid strongly coupled solutions a service-oriented architecture approach based on a neutral communication model and implemented with an appropriate middleware technology like web services is advised. This approach is called loosely coupled integration and offers a high degree of flexibility on utilizing the best-of-breed components to build the enterprise PLM solution. But even if such a solution may be called loosely coupled from an implementation point of view it requires the management of strongly related data models. Even if the underlying technical interface solution may survive a data model change, related processes need to be adapted. Additional attributes arising within one system and exchanged by loosely coupled interfaces need to be represented and understood by processes in the other system.

Collaborative and cross-enterprise PLM processes, where a centrally managed infrastructure is not available or not desirable, are characterized by the following aspects:

- A small number of participants collaborate in ad hoc manner without forming a permanent partner network.
- The isolated working processes are mature but not shared between the process owners. Interfaces exist, and they are established at all necessary levels.
- No changes through the lifetime of the collaboration are allowed.
- A single CAD system is in use at least for sharing.
- Legacy management tools like change management, document management or classification or configuration schemes need to be supported as they are. Every partner uses them on their own.
- No shared product management solution resides anywhere, elementary versioning and document tagging is in effect.

This approach is named *point-to-point integration* and is currently subject of research and investigation to support collaborative product development [15]. Without a central PLM solution and consequently without a single point of failure this solution is very flexible and robust. It is very easy to enter or to leave the design team. This approach suits the ad hoc nature of small-scale collaboration. This approach is accompanied by tagging technologies or utilizing ontologies as a common vocabulary to perform inter-project communication and to link digital product representation artifacts. Realization on top of peer-to-peer technology is an option but currently not supported by industry software applications.

16.6.1 PDM System Integration

16.6.1.1 Synchronous Integration

A synchronous PDM integration updates referenced PDM data more or less immediately between source and target systems. Whereby the actual update operation may consume significant computing resources it is expected to be very fast. In reality, synchronous update must be performed within several seconds. From an execution point of view a synchronous integration may cause the initiator of the update operation to stall until the operation has succeeded. But this depends on the implementation techniques used.

Common to all implementation approaches is the definition of a so-called trigger initiating the necessary update operation at the occurrence of a specified event. Current PLM systems provide extension points to define such triggers for instance in the database operation, within workflow or other business process operations, or as callback in the graphical user interface.

Whereby immediate update may be a requirement of the business for continuous flow of information, it may cause a large number of small changes to be executed. Within network based integrations this is not the optimum. Problems arise from the following scenarios:

- a significant number of changes is caused by automated processes like propagating date changes in effectivity control,
- interactive maintenance of PLM data may actually flip attributes,
- usage of wizards may cause interactive data cleaning,
- data maintenance may trigger other data updates which are handled better in combination, or
- volume of a single data set may be too large for a synchronous update to be performed without breaking existing timeout limits for blocking synchronous operations.

Transferring changes as fast as possible induces unnecessary load to mirror all changes in coupled systems. If history is not of concern, transferring state at defined times or after performing a well understood chain of interactive changes may reduce the data management effort. Furthermore, combining a large number of changes into a compound update data set may improve performance by reducing consumption of computing resources like network bandwidth and database updates. This is achieved by switching to asynchronous integration.

16.6.1.2 Asynchronous Integration

An asynchronous PDM integration updates referenced PDM data under the control of a third instance like an external timer or an external event signaling data availability. The actual update operation is controlled by the receiving system. Data to be processed by the update operation is already extracted from the source system and does not consume source system or network transport resources. The performance of the update in the target systems depends on the performance of the messaging and the receiving system only. Asynchronous integration decouples the system resources of source and target. This approach allows the aggregation of consecutive changes into a single change to the final stage if history is no concern. Keeping the order of changes the aggregated update can be performed much more efficiently by grouping update operations into a single target system operation.

For updates of large data sets like CAD data, asynchronous integration may be the only solution. If data sets comprise structure data that need to be efficiently processed, asynchronous integration may be performed practically online and the end user experience will not be different from synchronous integration but with better utilization of computing resources.

16.6.2 Cross-Domain Integration

The cross-domain integration addresses the challenges connecting different engineering disciplines like electrical, mechanical and software engineering. Whereby the disciplines do not differ in their notion of the general development process, the cycle schedule, the deliveries and their impact control, and the frequency of changes differ significantly. An integrated solution with a single system approach will fail in this situation. The integration needs to take into account.

- Isolated synchronization points only when a release or change exists during the product development and even during the whole lifespan of the product.
- Integrity of the solution is guaranteed by fulfilling interface contracts.
- Traceability (why a decision was taken) across domains is necessary to control the development.

The challenge of cross-domain integration is the replacement of the aspect of common data storage in a single database by the notion of information association across system and domain boundaries. The association between related information becomes additional external information. It is not sufficient to build up this relationship once and for ever. Whereby read access seems straightforward it is required to manage this association as an additional constraint, which needs to be taken into account for data access operations like update and delete. This additional reference information is used to realize the further use cases like where-used search, impact analysis, release management, which are illustrated below with their inherently increasing complexity. These functions make it easier to master the product creation process and avoid errors. This results in shorter development times and satisfies legal and customer requirements regarding traceability.

16.6.2.1 Where-Used Search

The where-used search identifies references to an asset in question by searching for all occurrences of a particular reference to that asset. The search may be narrowed to a subset of source item types. Narrowing the search to a particular reference depth may speed up the operation but will not fit in general to the use case which asks for instance like: “Given an item *X* in product *A* identify all other products using this item *X*.” This search is expensive in a single system context but it becomes extremely costly when executed across domains using different software systems. In this case is not sufficient to maintain cross-domain references in an additional data base but it is necessary to build up corresponding reference indices in the related software systems to support this use case efficiently. Depending on the chosen solution architecture the search may involve traversals in up to three databases, e.g. source, target and reference database, to succeed. Optimizing this search requires adaptions in the systems involved.

16.6.2.2 Impact Analysis

An impact analysis tries to find answers to questions like: “What impact will the changes needed to satisfy a requirement involve and which materials or software modules will be affected?” Other examples are: “Is there a test case for each requirement?”, “Is there a requirement for each material?” or “Is there a requirement and a test for each specified software function?” These questions are part of an audit support use case. Such use cases call-out for a traceability report which selects objects with an indication of all their dependencies and contexts of potentially different concepts which are not stored in the same database in general. These use cases are part of general methodologies like SPICE and CMMI and require the cross-domain references.

In contrast to the where-used use cases, it is not sufficient to traverse a product structure and find referenced artifacts of the same quality, e.g. item, product, but it is necessary to built-up and maintain associations between otherwise unrelated entities like parts, solutions, products, CAD files and requirements or software artifacts. This approach is easily applicable to the relevant consistence checks as mentioned above. The associations need to be volatile but revision-proof and may be built-up by classification attributes or semantic means.

16.6.2.3 Release Management

Mastering impact analysis is the precondition for automated maturity check on all modules and from all domains with the linking of the release level, requirements baselines, test results and much more. Whereby impact analysis may involve human interaction to select affected entities, it is desirable to select all necessary entities for release repeatedly, secure and automatically. Release management systems may maintain entire sets of affected entities to get hold of all induced changes. Whereby capturing this information in a single system is a well understood feature of PLM systems in conjunction with status management and workflow, integration with an external release management system involves the management of references across domains. The static part of this reference management is usually implemented with an external link database. Consistency and integrity, as well as workflow are challenges which need to be addressed by an integration solution.

16.6.2.4 Example: Integration of PDM with SDM/CAE

A special cross-domain integration scenario is to bridge the gap between design and simulation. Whereby some of the market leading PLM systems come with their simulation support inbuilt, like Siemens PLM Teamcenter and Dassault Systèmes V6, a number of proprietary simulation data management (SDM) solutions need external cross domain integration. Such SDM solutions like MSC SimManager, Altair Data Manager or ANSYS Engineering Knowledge Manager have similar functionalities like PDM systems to manage product information. A number of challenges [16] arise from the following requirements:

- Simulation data needs to be referable
  Simulation data are inherently redundantly used to evaluate alternatives and needs to be provided for the different simulation tools in their proprietary data and file format representation.
- Simulation data need to be organized within a product context
  Analysis is not restricted to a single part but always carried out under certain conditions for a purpose. PLM provides this context.
- Relationships with other domains need to be kept
  This challenge relates to the previous one because iterative changes in the design process in one domain cause respective changes in the other dependent domain which need to be evaluated and negotiated with respect to the validity of the assumptions.
- Simulation data needs to be managed during the product lifecycle
  Development in the interrelated domains is carried out concurrently. Synchronization occurs at predefined milestones or maturity gates. The corresponding lifecycle-state information needs to be managed.

With an integration platform it is possible to create an automated end-to-end process between the PLM world and such simulation data management systems

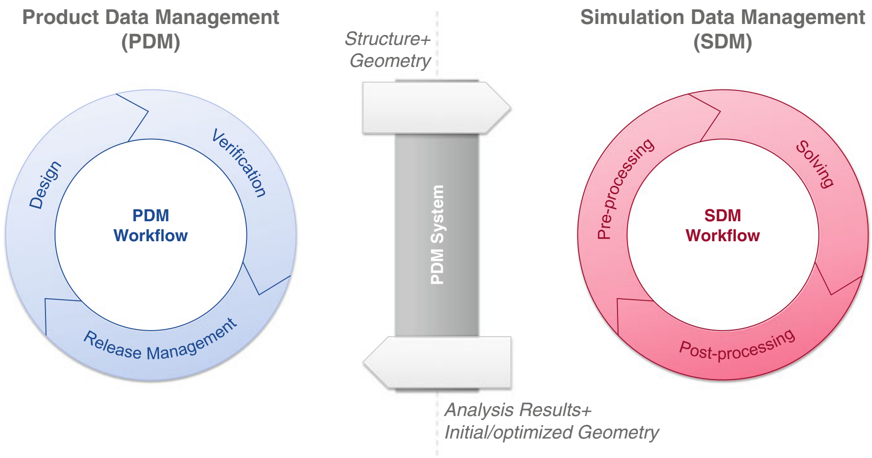


Fig. 16.3 General workflow of PDM-SDM integration

(Fig. 16.3). This process keeps track of changes on both sides, synchronizes shared data models and avoids additional manual work by making sure that all changes to relevant geometry, structure and connectivity are transferred between PDM and SDM systems. Furthermore, incremental update may reduce the volume of data so that it is perfectly suited for the simulation project. This requires filter mechanisms to configure the structure such that it is relevant to the SDM.

The integration platform supports the robust, fast and flexible integration of simulation and PLM processes by the following generic data transfer use cases from the PDM system to the SDM system:

- initial transfer of parts and assemblies including attributes and corresponding files (CAD and others),
- update of previously transferred data, and
- check for update in the SDM system, if changes to PDM data exist.
  This use cases supports the push communication model.

The following generic data transfer use cases are supported in the reverse direction from the SDM system to the PDM system:

- transfer of the data changed in the SDM system, e.g., CAD files or analysis results, back into the PDM system, and
- manually controlled data transfer from the SDM system to the PDM system.

Interactive filter use cases are necessary to reduce the transfer volume of data for a certain simulation project to just the parts required for that particular project. The PLM-related classification, configuration and status control mechanisms with their corresponding selection functionality are necessary in both domains. If they are not shared, a powerful mapping mechanism is essential.

16.6.3 Partner Integration

Especially for OEM another key question is: "How does collaboration with development and manufacturing partners fit into the PLM picture?" The level of integration into the PLM solution increases depending on the level of integration of the development partner from a part supplier up to a tier-1/2 supplier, which contributes with systems or to be completed products.

The partner integration can be operated in both a synchronous and an asynchronous way. The synchronous approach requires online access to data sources and systems and usually involves manual handling of data. This may be achieved by remote access to the PLM solution. This incurs security and infrastructure measurements to guarantee a safe and reliable mode of operations. A portal with associated data storage providing online access to data packages may be a cheaper option for online access.

A more automated, and therefore more reliable and repeatable mode of operation is offered by an asynchronous approach utilizing a data exchange tool managing data packages. Depending on the functionality this exchange tool may comprise basic data transfer functionality like FTP and ENGDAT, or more advanced features for data package handling, temporary storage and provisioning in a portal, or full-fledged PDM functionality to keep track of the exchange project separately.

A data exchange portal is well suited for distributing product data from the OEM to the development partner and re-integrating their deliverables. It is not always necessary, to distribute originally authored data. Simplified data, which do avoid an unnecessarily high data volume and ensure intellectual property protection of parameterized CAD data with full development history will be sufficient in most cases (see Sect. 18.6).

16.6.4 PDM System Migration

Introducing a PDM solution from scratch is the absolute exception. After more than 20 years of IT support some areas of PDM are already implemented with the help of IT technology. It does not matter whether the solution is based on a product solution or the solution is a legacy-based system. The data in this installation are of value for the enterprise and need to be migrated to the newly-chosen PDM solution. This requires a PDM system migration. Although migration can be seen as a singular activity synchronized with the PDM introduction process it often yields a situation in which migration forces a specific integration scenario.

In general, a system migration project follows the usual system introduction approach but source and target systems are usually known in advance. Nevertheless, several challenges exist. Those challenges arise from a process and from a source and a target system point of view at minimum.

16.6.4.1 Process Challenges

Introducing a new PDM system needs a business justification and usually addresses several shortages of the old solution. Improving the process support by a new IT solution requires adjustments to the original process at a minimum to leverage the PDM functionality to the actual processes. PDM system vendors try to minimize the impact of introducing their solutions by introducing a mature set of functionality. It is difficult to differentiate the market leading products by their functionality. Furthermore, PDM systems may be customized according to customer needs. Customization means implementing additional functionality in the PDM system. This may actually leverage the gap between the process in focus and the newly introduced solution. Unfortunately, this involves increased development costs, time, and resources at minimum, but lack of portability to new product versions at worst. The most difficult-to-handle problem is the lack of understanding how the new function would look like in the newly designed process. The potential end-user acceptance depends on the match between expectations on how the process will be supported by the new PDM solution while at the same time the process itself undergoes a transformation process. Therefore, it is necessary to implement a very sharp and visionary requirements management process which leads to a workable PDM solution and the same time leads the PDM customization process to reduce the negative impacts on portability and flexibility. PDM system vendors are usually not of much help in this situation. Their business driver is the product placement. Independent consultants tend to increase the customization efforts. It is the duty of the process owner himself to develop a workable vision of the processes in the new system.

From a process point of view this situation introduces two key questions to the migration: "Which data from the old solution will be required to drive the new process based on the new PDM solution?" And: "How do we need to restructure the existing data to accommodate the requirements of the newly introduced and customized product?" It will be clear from the statements above, that the answer to these questions controls the efforts needed in the system migration.

16.6.4.2 Source System Challenges

The source system serves two purposes during the migration: It is the source for feeding existing, well-known application data into the new system and it supports downstream processes, which consume the product data stored so far in the old system. Data representation in the source system must be fully understood and analyzed before the migration begins. This analysis is based on process and system know-how and needs to take into account implicit knowledge on history, legacy, and quality. The amount of data must be estimated. It might be an indicator for the migration costs, but is definitely a denominator for the business requirements of the migration schedule. Data clean-up and filtering before the actual transfer are necessary. It is in general not a good idea to propagate workarounds with respect to data quality or data consistency to the newly introduced system. Preparing the source system for migration is a perfect point in time to introduce a consistent terminology, a streamlined classification system, reduce ambiguities, and resolve historically motivated duplications or redundancies. The target system data representation capabilities will have an impact on data cleansing and on required data quality. This applies not only to product data but to CAD data in particular.

Even more complex is the management of downstream processes. If they are solely based on the original source system they need to be part of the PDM introductions process and they will undergo the same process transformation steps as described above. If downstream processes maintain their own data management and consume some of their data from the source system in question it is worth to justify the existence of this special solution. It might be an additional business driver for the PDM introduction process to incorporate this downstream process into the new target system to reduce the number of interfaces. At a minimum, downstream processes need read-only support until all necessary data are available in the newly introduced target system. This approach easily applies to read-only processes. It becomes more difficult, if downstream processes are not only based on but update the data source. Such processes need to be migrated in parallel to the PDM system migration. Their interfaces will evolve from existing interfaces to the source system and migrate to interfaces to the target system. These interfaces need to be developed in parallel. The business process will break and cause additional efforts, if they are not in place and functional by the first time of migration. Requirements for backward migration will be the consequence. This might cause the extension from a unidirectional migration scenario to a bidirectional integration scenario. The migration approach needs to be flexible to this requirement. This is especially the case, if the migration is part of a transformation process and the impact of all business drivers is not known in advance.

16.6.4.3 Target System Challenges

Ideally, for a successful migration the target system is up and running stable for a while. Access control is established. The end users are trained in the new system and they are aware of the new features of the new data model and the processes implemented within the target system. This approach is the perfect setup for a migration project to begin with. The target is well-understood and ready to use.

In practice, this will be the case for a small fraction of the migration projects. Usually, the target system is under development, the data structures are still changing and even more important, customer requirements are changing and customer expectations will develop as soon as customer data are visible in the context of the target system. The approach becomes more complicated as soon as customer acceptance depends on availability of live customer data in the target system. Migration needs to be prepared and performed at least partially in parallel to the overall PDM introduction process. This causes high pressure on the flexibility and speed of the migration approach.

All know-how aspects of the data model which apply to the source system as described above apply to the target system as well with one exception—history and implicit know-how are reduced to a minimum. It is a challenge to define future-proof stable mappings between source and target data models. We find no difference with respect to complexity, completeness, consistency and integrity between mappings for integration and migration. Of course, a clear understanding and expectation on data quality and data cleaning will help to reduce the effort. Migration needs to maintain data quality.

16.6.4.4 Migration Project Approach

The migration project is divided into the well-known phases of software development: requirements analysis and specification, design and build, acceptance tests and productive go-live (Fig. 16.4). The acceptance test phase differs from the common software development cycle. It splits into a phase of small batch tests which proofs the implementation to be compliant with specification and requirements and a test phase, which may be performed repeatedly or which may iterate over improvements of the specification, implementation and test phases. The tests try to identify the impact of the implemented migration solution on the complete source data set considered for migration. This approach requires a full dump of the source data to a separate stage solely dedicated to the test phase. This step proofs the assumption of the requirements phase to be correct and helps to identify anomalies or unexpected data in the source database. Thus, all data considered for productive migration need to be taken into account. The user acceptance test is

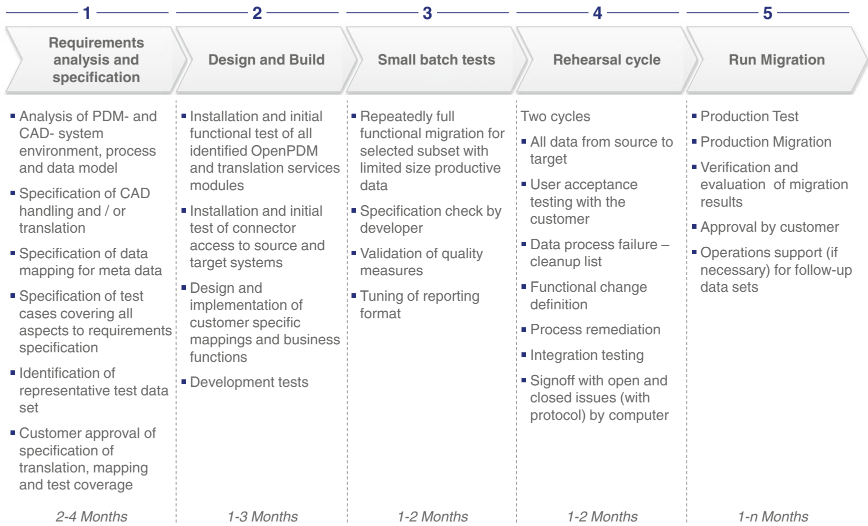


Fig. 16.4 Migration project plan

performed on the complete data set in advance and not just on a well-chosen subset. The data processing failures are logged in a cleanup list, which will be evaluated from a business perspective. Not all defects necessarily need to be fixed. Depending on the business value of the identified problems user acceptance may be achieved simply by ignoring a particular error condition, fixing the problem in the source data or excluding data from migration. The expected outcome of this phase is that there is no doubt on the result of the migration. Risks are reduced to a minimum and are isolated to runtime problems like system and resource availability.

Furthermore, the test phase gives an indication of the results to be expected from a performance and a reporting point of view. Both aspects need to be tuned to the customer's expectations before the productive migration takes place. Especially the performance needs to be managed because it drives the scheduling of the migration process and proves an optional splitting approach to be valid or not.

Depending on the nature of the migration, the go-live may be a singular event, which is a very rare case, or it may take several steps due to the chosen migration strategy. To reduce the risk associated with the migration or to split the overall effort in manageable sub tasks a step-by-step approach is more appropriate. The steps shall be derived from business drivers like products, programs, or subdivision. It needs to consider basic preparatory work, common data like standard parts or libraries, and will handle structural data separately from mass CAD data.

Depending on the requirements of the processes based on the source system and the chosen splitting approach a partial re-synchronization may be necessary (Fig. 16.5). This involves an additional data flow back from target to source. The data flow back asks for an automated solution which tends to increase the complexity of the uni-directional migration to a bi-directional integration scenario.

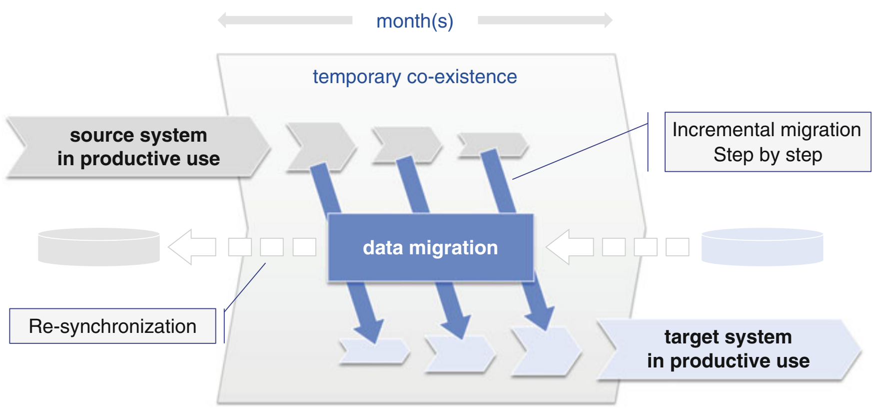


Fig. 16.5 Migration with temporary co-existence

16.7 The PLM Integration Platform

A PLM integration platform provides a common, easily available access mechanism to all necessary PLM information for collaborative engineering in a heterogeneous system world. Standardized interfaces to the majority of the required PDM functionality allow short set-up times for establishing working partnerships. Such a PLM integration platform is a product designed for the exchange of PLM information between different in-house systems, cross-domain integrations as well as partner data exchange. All components and processes are optimized for PLM structure handling. The platform enables to access the PLM data located in different systems via a standardized interface. Those interfaces with a wide range of PLM systems are realized in specialized connectors. As a base functionality these connectors can read and write objects, relations, files and of course all corresponding attributes (Fig. 16.6).

The PLM integration platform build more coarse-grain functionality by orchestrating and combining the logic to read and write PLM structured data by using the generic fine granular interface of the PLM system connectors. As a result, the export and the import of complete PLM structures together with the referenced secondary content like CAD, which is the primary data for the design engineer in almost all cases, are managed by such an integration platform.

The user interface, if necessary, is provided by a direct PLM system integration. This integration is controlled directly within the graphical user interface of the PLM system and gives the user the opportunity to select and filter information without changing the user interface. As a result, the direct integration communicates with the PLM system on one side to collect the data and with the integration platform on the other side to collect the necessary receiver information. The receiver information is needed for routing purposes but as well for filtering or transformation

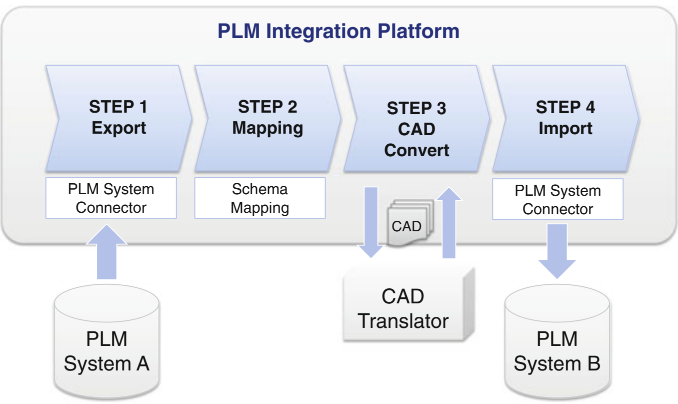


Fig. 16.6 Integration platform with workflow ⇒ PLM system connector ⇒ PLM system

control. Finally, the sending PLM system usually manages a record for all data to be exchanged to allow efficient retry or control, and for bookkeeping.

16.7.1 Cross-System Integration Scenario

A PLM integration platform is able to implement aggregated simultaneous views on multiple PLM systems. Sharing a common data model as defined by the specification and merging of multiple data sources into such a common view is an obvious, but complex application. The integration platform provides not only a shared and system-independent view on PLM data, but defines the semantics of access functions and the necessary answers of the connected PLM systems to form valid PLM data. The aggregated view on multiple PLM systems is in reality a one system-like image on distributed data. Users are no longer forced to use multiple user interfaces and multiple terminologies to access similar data from different sources. This shared representation requires structural, semantic and timely transformations. The platform provides such transformations which are based on a well-established transformation chain from PLM system specific data to generalized PLM representation, and to the special purpose end user representation.

Furthermore, the integration platform with its generalized data model allows an easy replacement of product data viewers to meet the requirements of different user groups. Based on the generalized system independent PLM data representation specialized viewing and processing solutions will support designers and other end users directly. If the cross-system scenario is based on a synchronous data access solution and all data access are realized online the need for erroneous data replication is minimized and applications may work on original data. This will allow optimistic change strategies to be implemented but may cause complicated interlocking scenarios which are not easy to manage or will cause inefficient data locking behaviors when multiple changes compete in the process.

16.7.2 Cross-Technology Integration Scenario

Instead of binding the integration solution directly to an implementation data model of one of the involved PLM vendors, a neutral representation based on the model-driven architecture (MDA [17]) approach is more suitable. Within the generalized MDA approach the necessary transformation steps from the vendor specific representation to the neutral, potentially standardized representation are defined. Utilizing well-defined data models (UML [18], XMI [19]) XML representations can be easily derived. Such examples are the IBM PLM framework infrastructure [20], the OSLC [21] infrastructure or the PLM Services [22] set of specifications. Bindings to programming languages exist as standards or quasi-standard industry solutions. Thus, a seamless transformation between an XML message comprising a PLM data set of information and a specific language or technology binding exists. The MDA nature of the data representation specification would allow the extension of the data exchange capabilities to new implementation techniques by defining a platforms specific mapping without losing the semantic meaning, but gaining the flexibility of multiple language or representation bindings.

One important way of data exchange in PLM is the STEP file representation. The PLM integration platform is suitable to implement integration scenarios with existing STEP file-based infrastructures. A complete STEP Part21 representation suitable for processing by industry strength STEP processor is a must. A lot of processes, especially in aerospace and defense, depend on such implementations. Due to the normative mappings between AP214 [23] AIM, which is the data model exchanged in STEP files, and the XML representation, exchanged by the Web Services defined by the integration platform interfaces, a complete round-trip and unambiguous data transfer scenario becomes feasible.

16.7.3 Cross-Domain Integration Scenario

The PLM integration platform is suited to implement cross-domain scenarios interconnecting PLM, CAD, ERP and other planning system data sources. The structural information provided by product or document information given in the PLM system provides a comprehensive access methodology to complete product definition data including support measurements for manufacturing, process data, logistics, change management and others.

16.7.4 Cross-Company Collaboration Scenario

Highly complex manufacturing goods like the products of the automotive, aerospace or consumer goods industry today need a close collaboration between the original equipment manufacturer (OEM) and its suppliers within an integrated context (Fig. 16.7). There is a strong need to collaborate during the design phase of a product and exchange information between partners (see also Chaps. 7 and 8).

Organizational data are usually within the scope of the specification of the connected systems. All PLM information may be associated with their corresponding creator and owner information, although it is not obvious for external resources. They need special consideration mainly for two reasons: (1) management of external resource information requires special efforts, and (2) external resources demand competitor safety, so that multiple external resources need to be granted access to carefully chosen subsets of information to avoid conflicts with non-disclosure agreements. Thus, the utilization of alias names suitable for cross-company referencing of parts, documents and others might be a solution. A better approach is the separation of all external bodies to prevent unnecessary cross-company access or

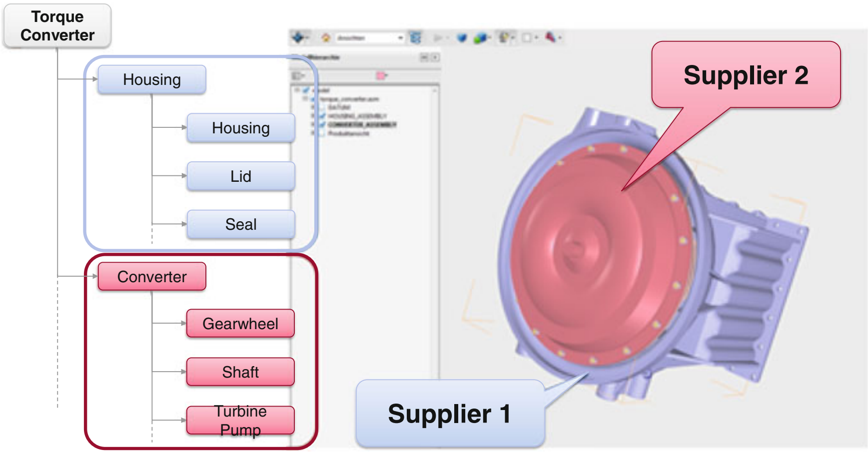


Fig. 16.7 Assembly with parts highlighted which are designed by suppliers

communication and the delegation of resource management to the external partner directly. With the corresponding legal and information management scheme in place a collaboration partner is able to allow its employees access to data and to guarantee that access control information is current and accurate.

The scenario is realized by providing a definition for a common generalized neutral data model and access methods based on common web technology, easily accessible for a large number of automotive companies. It is state-of-the-art to use a web-based transport mechanism which delegates aspects of authentication and authorization as well as encryption and other security measurements to already existing technologies and infrastructures. The PLM integration platform just utilizes this framework to access valid PLM data within user or system context. The framework comprises the aspects of generalized neutral data representation, data mapping, and atomic data handling functions. The data handling can be easily combined into more complex activities, which are controlled by a workflow mechanism and a framework for access control and process management. The framework solution is accompanied with extended logging and monitoring facilities. These services are highly customizable to meet a wide range of customer demands and to be integrated in the management and reporting services. This allows the build-up of efficient interactive scenarios.

16.8 Case Studies

The following case studies comprise realizations of the use cases described above. They gather our experiences from real-life applications built with the software products described in Sect. 16.7. We introduce the details and aspects of those solutions to the extent of understandability and comprehension.

16.8.1 Automotive

The central PLM concept in the automotive industry is the platform approach which is currently extended into a modular approach. Several car lines of an automotive OEM have an identical platform or are built based on the same module set sharing parts, components, concepts, manufacturing principles, supply chains, maintenance procedures and so on. For the OEM this leads to a reduction of costs on the one hand due to a standardization and re-use. On the other hand its modularization is the key to an increasing variety of products. This variety needs to be managed, approved and documented. This also leads to a very complex PLM concept due to the necessary modularization of PLM structures and the management of product changes over a larger number of car derivatives (see Sect. 21.2).

Since several years a cross-OEM partnership for the design and the production of several vehicle types is established. From the PLM point of view two types of partnership are established: collaboration and asynchronous integration. The collaboration approach is typically used for long term partnerships or will be reused for several project scenarios. The asynchronous integration is well known for ad hoc partnerships and short term or single project partnerships.

16.8.1.1 Collaboration

Focus of a collaborative partnership is the development of a cross-company platform concept. Both partners build this platform contributing their product expertise. To support this collaboration with PLM concepts, typically a central PLM system is used to manage data of both partners. There is no need for all partners to internally run the same PLM solution as the so-called Partner Hub. It is an independent solution with a clear separation from the internal systems. Editing PLM data is typically only carried out in the internal PLM system. This avoids the situation that a designer has to work in two different PLM environments. The Partner Hub mirrors only the PLM data that are partner relevant.

From a PLM process perspective the challenge is to provide in the Partner Hub only such data that are needed to fulfill the collaboration with the partner. In addition, data need to be synchronized with the local PLM system in an automatic way to avoid replicated work and version mismatches. To establish such collaboration, internal data are typically pushed to the Partner Hub initiated by triggers. The trigger is fired at well-defined stages of the design process in the internal PLM system to synchronize the PLM data from with the Partner Hub. This connection is typically established by using a synchronization framework with connectors to the internal PLM system and to the Partner Hub.

Technically this synchronization acts like the PLM integration described in Sect. 16.6.1, but of course with a much more complex logic to secure the process and to avoid the exchange of unauthorized data. The most significant difference is the location of the Partner Hub in the demilitarized network zone of one partner or on an external server. After the trigger has initiated the data transfer from the internal PLM system to the Partner Hub, the synchronization framework exports all related data out of the internal system and maps the data structure to the scheme of the Partner Hub. CAD data need a special focus before the transfer to the Partner Hub is started, because not all content of the native CAD data should be published to partners. Sometimes just neutral formats like JT will be provided or native CAD data are prepared as described in Sect. 18.6.3.2. Only these processed CAD data are transferred to the Partner Hub.

From usage perspective there are two concepts to view and work with data published in the Partner Hub. The simplest way is to use the standard PLM system client to access the Partner Hub to view or download the data interactively. The second way is to download and import the data automatically into the own PLM system. With the help of effective PLM selection operations and lightweight viewing facilities capable of web-based geometry inspection the design engineer identifies the relevant changes in the structure and defines the geometry models that have to be exchanged. After that the engineer initiates a data exchange process to transfer the identified data to the own internal PLM system. The exchange process keeps track of all the interactions, converts CAD models according to the guidelines of the company exchange processes and imports the data in the internal PLM system according to the defined data mappings. The partner data can now be used in the standard internal PLM processes.

This real-life example demonstrates the capability of the integration platform. It allows the interaction of several tools on a neutral and open foundation, namely the PDM system providing the necessary structure information, the exchange infrastructure providing the data converting and messaging capabilities and the web browser technology providing easy access to cross-company and cross-domain functionality. Every single building block of this tool set is exchangeable with another one compliant to the standard interfaces. The product specification is capable to support the required functionality under productive conditions.

16.8.1.2 Asynchronous Partner Integration

Another common approach for partner data exchange in the automotive industry is the Asynchronous Partner Integration. This approach uses a data exchange tool to transfer data from one partner to the other and does not define a central collaboration management tool. This approach defines high demands on the data format and the package sequence during the exchange.

Different to a collaboration platform in which a central tool provides information to all partners and data exchange is supported by a synchronization framework, a neutral data model is a central aspect for asynchronous data exchange. In the automotive industry typically standards like STEP [23] or JT [24] are used for data exchange (see Sects. 21.6.2 and 21.7). It is the responsibility of all partners to provide and accept data in the defined neutral format. PLM integration platforms are typically used to support the data import and export scenarios.

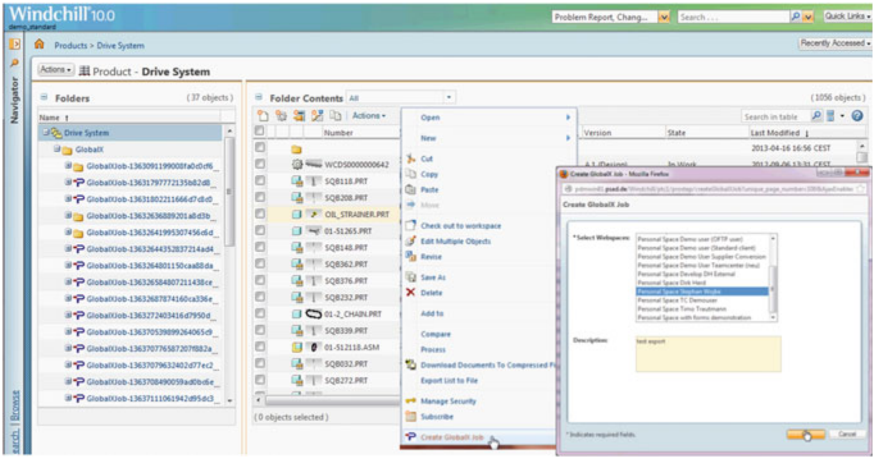


Fig. 16.8 Collaboration platform realized with Windchill

Our implementation with OpenPDM and OpenDXM GlobalX provides a new callback in the graphical user interface in the PLM system. A user interactively selects product data by choosing a root element and applying some filtered traversal and selection operations. Then, he starts in the PLM system the data transfer. Figure 16.8 shows the data selection and the partner data transfer dialog fired from the graphical user interface of the PLM platform Windchill and communicating with the data exchange platform.

The integration platform starts the background process to export the selected data and maps the data to a neutral format. It yields a package of metadata with linked CAD data in the neutral formats. This package will be handed over to the data exchange tool, which performs the transfer to the partner. The partner system recognizes the new package and starts an import process with its own integration tools. To avoid corrupt data in the target system, the import engineer uses tools to perform a dry run import. Based on visual inspection, he can decide to import the data or to decline this data delivery. A central aspect is to guarantee the sequence of data package imports in case of huge volume. The export, transfer and import of the PLM data from one partner to the other can take more than one day. In result it is possible that small packages can overtake big packages in the data transfer process. If the package sequence is not recognized during the import the loss of data can occur. The data exchange tool prevents the lost update in that scenario.

16.8.2 Aerospace

The aerospace industry heavily depends on stringent documentation of their product along the whole lifespan. Where the number of products is significantly smaller than in automotive, the sheer volume of product data, e.g. number of parts, number of documents, variety of configuration options, is much larger. Even the lifespan of the product is longer and usually challenges the lifespan of IT systems that manage the design process [25]. The documentation needs to store information on the complete product structure "as-designed", "as-defined", "as-planned" and "as-maintained" and not only as reference but as individual representation of every single aircraft identifiable by its serial number.

16.8.2.1 PDM-ERP Integration

Both PDM and ERP systems manage product-related data. Common denominators are part master data, configuration and effectivity, but product structure differs. PDM systems manage highly detailed design structures. This view on product structure is the engineering bill-of-material and feeds the bill-of-material in the ERP view. ERP stores the planning bill-of-material which manages manufacturing or maintenance planning. Its structure supports assembly processes, logistics, and maintenance or spare part management in after-sales processes. This structure reflects not only the design hierarchy, but the product management hierarchy (Fig. 16.9).

The PDM view evolves during the development and change phases of the product. This view will change if design principles change. Their lifespan is significantly shorter than the manufacturing and maintenance phase of the ERP structure. The planning structure is also significantly flatter than the design structure.

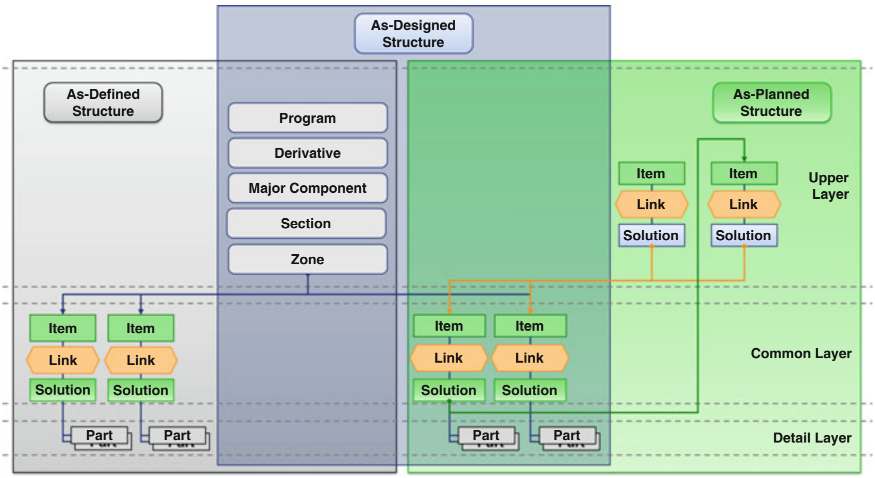


Fig. 16.9 Elements of definition, design and planning product structure layers

16.8.3 Consumer Electronics

The consumer electronics market has an extremely short time-to-market. The lifespan of devices is significantly shorter than in automotive or aerospace. The number of players is larger compared to the automotive or aerospace industry. Customer expectations are targeted on usability, energy efficiency and increasingly on sustainability. Success in such a mass market strongly depends on a vendor's ability to react fast on trends, to be cost efficient at the same time and to deliver well-designed gadgets. New trends in mobile and connected solutions demand constantly innovations. Simulation of mechanical and electro-magnetic properties is an essential part of the development process and a must to choose alternatives for a constantly changing product design. The integration of PLM with simulation data management systems is an application for cross-domain integration.

16.8.3.1 PDM-SDM Integration

With OpenPDM an integration of PLM with SDM was built to allow an extremely short lead time to incorporate simulation analysts into a team of designers and to work efficiently co-located. Thus, it was planned to setup an additional project environment within the PLM system which stores all necessary data of the original product structure as shallow copy. This project specific workspace is referenced from the SDM system. Analysis is performed on propagated information of the reference data set. The functionality of the integration comprises these use cases:

- Setup of a reference project environment
- Initial load of a selected reference substructure into the SDM system
- Refresh the initially loaded substructure

The refresh is carried out on the main assembly and all its children to propagate structural changes and releases, and to monitor release state and design changes in the SDM system workspace (Fig. 16.10).

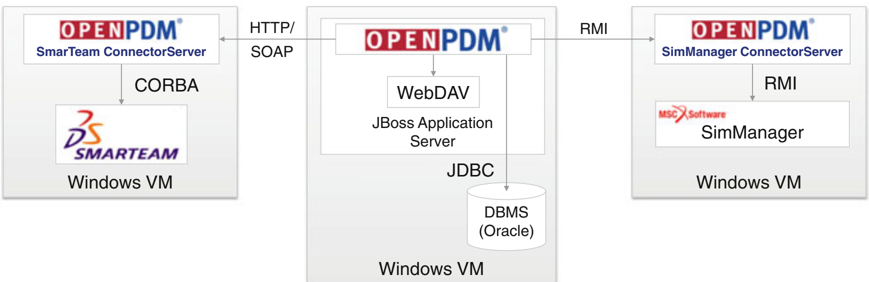


Fig. 16.10 PDM-SDM integration architecture

The refresh functionality was supported both by push and pull mechanisms. The PLM system could trigger the refresh function to push changes to the SDM system. The SDM system is capable to pull changes from the PLM system. This was bound to a user interaction. The cross-domain communication can be initially supported by those basic building blocks. But this elementary approach leaves some potential for further improvement like

- Feeding back of simulation results to the PLM project
- Documentation of design quality checks
- Induced model changes in the simulation need to be reflected in design domain.
- Handling of variant and alternative design data.

16.9 Conclusions and Outlook

PLM is a complex concept to deal with the challenges in state-of-the-art product development. The PLM IT solution is just a single constituent. The IT solution itself consists of a number of integrated tools. The integration scenarios show common patterns but differ in detail and level of interaction.

Central to PLM integration is a common set of product structure information. Every lifecycle stage utilizes its special aspects of product structure. The representations as-designed, as-planned, as-build are just examples. Their functional relationship is defined by the use cases they are applied to. The use cases have their real-life counterpart in business processes. The business processes are specific to the industry. To some extent they are differentiators of the market players. The implemented PLM solution is not a decisive advantage of competitors. Its level of business process support definitely is. It is questionable, if COTS solutions "one size fits all" will meet all process requirements. More generic template approaches addressing the needs of standardized processes will ease adaptation but will never avoid customization. Furthermore, it is not clear, if harmonization and standardization of processes is desirable across an industry to accommodate COTS utilization. Today, specific software solutions support the product development process in practice. It is questionable, if all lifecycle stages of a product shall adapt to the same business processes which has proven to be successful in a single business.

The level of business support may be judged by simple evaluation of the product lifecycle solution and its supporting measures. Taking into account, that PLM solutions are usually integrated solutions, the biggest advantage of an implemented PLM concept arises from the seamless flow of information across all boundaries: system, domain, technology and enterprise. We have illustrated how a specialized integration tool may help to bridge those boundaries. With the presented toolset a wide range of requirements was accommodated. New PLM components and application can be easily integrated without breaking existing integrations. The integration approach is customizable along the development and customization path of the connected components. The integration framework specialized to PLM concepts has a significant added value in comparison with enterprise service bus solutions or EAI solutions. The PLM integration framework is specialized to PLM concepts with modules to create, read, update the product structure together with well-designed building blocks to support PLM patterns like versioning, check-in and check-out, and configuration. The general purpose of enterprise service bus solutions is bridging technological gaps, while EAI solutions focusing on the exchange of tabular data, which are a subset only of the structured product data.

New challenges arise from the demand for better support of the PLM-CAD integration and from new technologies like mobile devices or computer graphic technologies asking for new metaphors to handle PLM data, not by browsing data columns but by exploring graphical representations of the product with a selected set of PLM data as overlay, potentially in an immersive graphical environment.

References

1. Eigner M, Stelzer R (2009) Product lifecycle management. Springer, Dordrecht, Heidelberg, London, New York
2. Abramovici M, Schulte S, Leszinski C (2005) Best practice strategien für die einführung von product lifecycle management. Ind Manage 21(2):47–50
3. Tavcar J, Potocnik U, Duhovnik J (2013) PLM used as a backbone for concurrent engineering in supply chain. In: Stjepandic J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Springer, London, pp 681–692
4. Katzenbach A, Steiert H (2011) Engineering IT in der Automobilindustrie—Wege in die Zukunft. Informatik-Spektrum 34(1):7–19
5. Silcher S, Seeberg B, Zahn E, Mitschang B (2013) A holistic management model for manufacturing companies and related IT support. Proc CIRP 7:175–180
6. Schuh G (2006) Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte. Springer, Berlin, Heidelberg
7. Vajna S, Weber C, Bley H, Zeman K (2009) Integrated design engineering. Springer, Berlin
8. Stark J (2011) Product lifecycle management. Springer, London
9. Dekkers R, Chang C, Kreutzfeldt J (2013) The interface between "product design and engineering" and manufacturing: a review of the literature and empirical evidence. Int J Prod Econ 144:316–333
10. Abramovici M, Bellalouna F, Neubach M (2010) Delphi-Studie PLM 2020. Ind Manage 26 (3):47–50
11. Abramovici M, Bellalouna F, Flohr M (2008) PLM für individuelle reale Produkte. Ind Manage 24(3):41–44
12. Sendler U (2009) Das PLM-Kompendium: Referenzbuch des Produkt-Lebenszyklus-Managements. Springer, Berlin, Heidelberg
13. Schuh G, Assmus D, Zancul E (2006) Product structuring—the core discipline of product lifecycle management. In: 13th CIRP international conference on lifecycle engineering, pp 393–398
14. Mechlinski T (2007) Aktuelle Konzepte für die PDM-Integration. Produkt Daten Journal 14 (1):34–37
15. Stiefel P (2010) Eine dezentrale Informations- und Kollaborationsarchitektur für die unternehmensübergreifende Produktentwicklung. PhD thesis, Technische Universität Clausthal, Clausthal-Zellerfeld

16. Grau M, Trautmann T (2007) Vendor-independent integration of CAD and CAE processes based on OMG PLM Services. NAFEMS 2006
17. OMG (2003) Model driven architecture guide. http://www.omg.org/cgi-bin/doc?omg/03-06-01. Accessed 23 Aug 2014
18. OMG (2004) Unified modeling language (UML), ISO/IEC 19501. http://www.omg.org/cgi-bin/doc?formal/2004-04-04. Accessed 23 Aug 2014
19. OMG (2005) XML metadata interchange (XMI). http://www.omg.org/cgi-bin/doc?formal/2005-05-01. Accessed 23 Aug 2014
20. Credle R (2008) SOA approach to enterprise integration for product lifecycle management. IBM, International Technical Support Organization, Raleigh
21. OSLC (2013) Open Services for Lifecycle Collaboration Change Management Specification Version 2.0, http://open-services.net/bin/view/Main/CmSpecificationV2. Accessed 23 Aug 2014
22. OMG (2011) Product lifecycle management (PLM) services. http://www.omg.org/spec/PLM/2.1. Accessed 23 Aug 2014
23. ISO (2003) STEP 10303:214 core data for automotive mechanical design processes
24. ISO (2012) 14306: Industrial automation systems and integration—JT file format specification for 3D visualization
25. Mas F, Menendez J, Oliva M, Rios J (2013) Collaborative engineering: an airbus case study. In: The manufacturing engineering society international conference, MESIC. Elsevier, pp 336–345

Chapter 17
Variability Management

Georg Rock, Karsten Theis and Patrick Wischnewski

Abstract The global market, different and changing environmental laws, the customer wish for individualization, time-to-market, product costs, and the pressure on manufacturers to discover new product niches, to name only a few variability drivers, result in an ever increasing number of product variants in nearly all engineering disciplines as, for example, in car manufacturing. Mastering the related increasing product complexity throughout the whole product lifecycle is and remains one of the key advantages in competition for the future. Currently for a manufacturer, as for any other discipline, it is no option not to invest in an efficient and effective variability handling machinery able to cope with the arising challenges. Not only the task to invent, develop, introduce and manage new variants is important but also to decide which variant to develop, which to remove and which to not develop at all. The consequences of such decisions with respect to product-line variability have to be computed based on formalized bases such that an optimized product variability can assure on the one hand customer satisfaction and on the other hand cost reduction within the variability-related engineering processes. This chapter presents current research in the field of product variability configuration, analysis and visualisation. It presents solution sketches based on formal logic that were illustrated by some real world examples.

Keywords Product variety · Mass customisation · Product configuration · Complexity management · Variability management · Formal variability analysis

G. Rock (✉)
Trier University of Applied Sciences, 54293 Trier, Germany
e-mail: g.rock@hochschule-trier.de

K. Theis
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: karsten.theis@prostep.com

P. Wischnewski
Logic4Business GmbH, 66123 Saarbrücken, Germany
e-mail: patrick.wischnewski@logic4business.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_17

17.1 Introduction

Product variety arises because customer requirements are usually individual and therefore they expect an individual solution for a given investment. Private consumers rank variety of assortment straight after location and price when naming reasons why they shop at their favorite stores. They care about variety because they are more likely to find what they want when going to a store that offers more varied assortments. In a market which is predominantly determined by the customer ("buyer market") and subject to high volatility, the variety of products for reasons of competition is presumably imperative because every manufacturer is forced to continuously extend the product portfolio with new, more efficient, more attractive products. Furthermore, the product variety occurs on the long-lived capital-intensive goods too, which were brought on the market to an earlier time with a lower level of modularity and now need to be overhauled or upgraded.

Thus, customers get a wide product portfolio offered, which allows them to find a product that best meets their requirements. The challenge lies in the selection of products, less the actual product offering [1]. Consumer demands have increased significantly in recent years and will increase again. Enterprises become successful if they can provide solutions for a variety of needs. The need to offer a variety of products marks the crucial difference with the past. From the marketing perspective, vendors offer product variety presuming that the product characteristics determine their value and that variety is a key driver of utility. Their assumption is that customers derive utility by choosing among singular characteristics of a product. Product variety can be just unnecessary, if the vendor is not able to meet customer's needs producing the most appropriate product from an existing modular product platform (see Chap. 14) [2, 3].

The use of information and communication technology in modern cars gives the product managers an even increased and thus arbitrary range of variety [4]. Especially the entertainment systems in automotive industry will be revolutionized concerning functionality and usability aspects. The seamless integration of mobile systems and even the utilization of mobile platform technologies within the car's entertainment system itself will result in new variants that have to be handled correctly in the car manufacturing context.

In a future perspective for the automotive industry, this trend of increasing variability is expected to be reinforced [1]. The automotive industry will meet a world of consumers in the future who want to get their individual wishes, needs, expectations and preferences expressed by their automobile and simultaneously the customer expects an improved product quality. Their individuality comprises various sub-areas such as mobility, urbanity, emotions, entertainment, security or knowledge. All these aspects can be condensed into the forecast that the passenger car in the future will be an expression of one's personality even more than today. It must, therefore, be better adjusted to the individuals than ever before [5].

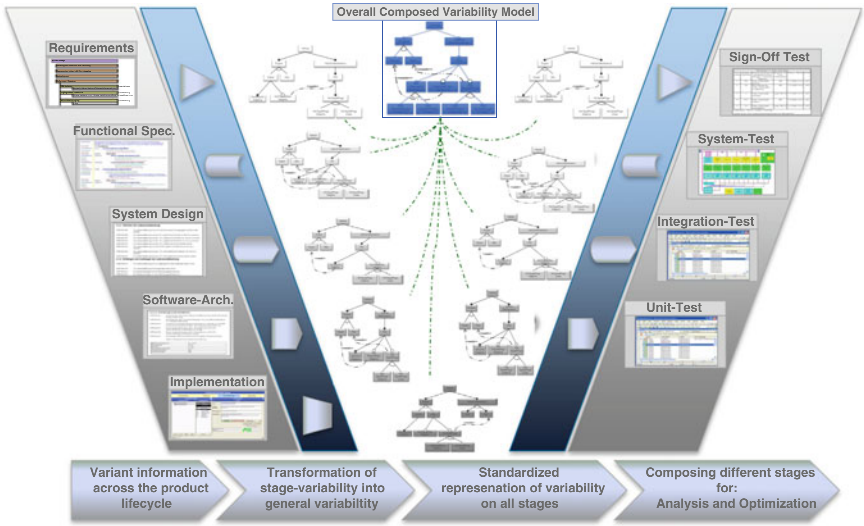


Fig. 17.1 Variability as a cross-domain and cross-functional development characteristic

Changing the view from a user of products to a developer of variant products and focusing on a V-Model based development the variability problem arises in every development stage as illustrated in Fig. 17.1 (see also Chap. 9).

Starting at the very early phase as for example in requirements engineering (see Chap. 5) variability constraints arise in all successive phases, as there are design, construction, implementation, and during the complete validation or verification phase. This shows two main aspects of variability in development (so-called internal variability). First, variability is not limited to exactly one development phase, but it rather spreads over all development phases. Second, since there are dependencies between domains and corresponding functions we have to have an "overall composed variability model" as depicted in Fig. 17.1. This model tends to be very large and complex. Thus, there must be a very powerful methodology and corresponding analysis engines supporting the engineers during product development.

The structure of this chapter is as follows. In Sect. 17.2 we present some background information including a short introduction to the current research in variability management. Section 17.3 changes the perspective from a user of variable products to the developer of products with variability. The related development process, challenges and corresponding concepts for a tool-based solution are presented. In the following Sect. 17.4 the application of the described concepts is shown with the help of industrial examples. Although these examples are rather abstract they illustrate typical application scenarios. Section 17.5 presents use cases. Two final Sects. (17.6 and 17.7) sum up and give some future directions in the field of variability management and variability analysis.

17.2 Background

Product variety encompasses different product designs or product types that are offered in a market by a vendor. It can be subdivided into external variety and internal variety. External variety is the product variety seen and perceived by the customer, whereas internal variety is the variety covering all procedural variants inside manufacturing, such as logistics and distribution operations in satisfying the provision of external variety. External variety is further subdivided into useful variety and useless variety. Useful variety is appreciated by the customer, such as useful options and stylistic differences, in distinction to the useless variety that is either transparent or unimportant, or confuses the customer. Uncontrolled internal variety may yield the excessive and unnecessary variety of parts, features, tools, fixtures, raw materials, and processes. Although more product variety extends the potential for generating increased revenues, there are potential adverse effects, resulting from increased complexity.

External variety is an important driver for enterprises producing to forecast. On the other hand management of internal variety is dominant for enterprises offering products to order. The effectiveness of strategies to mitigate negative effects of internal variety, such as modularity, mutability, late configuration (postponement), and option bundling, depends on the order-fulfillment strategy the enterprise follows [1]. This requires that assembly systems have to be designed such that they can handle the variety, too [6–8].

Hence, the main objective of many recent research projects is the decrease of external variety by standardized interfaces and further module utilization in different combinations and simultaneously the limitation of internal variety by standardization and modular design. In addition, research has focused on the identification of the right degree of variety [9–11]. This aims at reducing the complexity such that it can be handled with current tools [11].

Now, the question arises how the remaining internal product variety can be managed such that the following requirements are simultaneously fulfilled:

- As many customer requirements as possible are fulfilled,
- the quality of the products is improved,
- the development and production time is shortened, and
- the overall costs are reduced.

Since product variety has gained a high importance in the past years, a new, autonomous cross-functional discipline called variety, variant, or variability management was established for holistic treatment of these phenomena. In particular, powerful tools that shift dealing with the complexity from the user to the tool, is, in our opinion, the key for successfully achieving the above requirements.

The feature oriented domain analysis-approach (FODA) introduced by Kang [12] uses feature models which are today almost established as a standard mean to specify the variability of products. They allow for a description of the domain engineering and the application engineering model variability within the Product-Line-Engineering

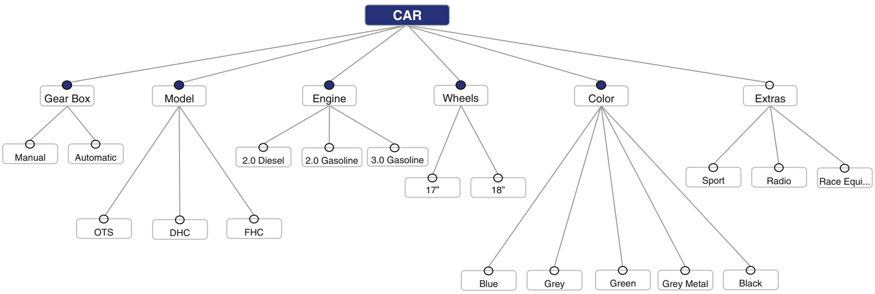


Fig. 17.2 Product structure

process as described in [13]. Although feature models have not yet pervaded the complete market for variant reach products, we observed over the last years that companies using their proprietary formats and analyzing techniques (often based on Excel-like tools) reach the limits of their approaches and are looking for a more sophisticated analyzing machinery and a standardized form of variability specification as started for example by the Object Management Group (OMG) Request for Proposal for common variability language (CVL) [14]. Feature models enable us to visualize variability in multiple ways as shown in [15–19]. All the referenced approaches are using tree like views as for example shown in Fig. 17.2. Furthermore they give us the possibility to analyze the specified variability in a formal way because of the defined and generally accepted formal semantics as described for example in [20–22]. There are several extensions of these models including the handling of additional feature attributes to be part of the formal analysis process as described in Sect. 17.3. At least from an academic point of view variability related problems in product development seem to be solved and do not need further investigations [23]. A closer look at real world applications reveals that many of the problems remain still unsolved for different reasons. A short and by far not complete list of reasons is as follows:

- The pure size (still increasing) of real world problems,
- Missing formal variability experts able to improve formal analysis algorithms on a problem/product/enterprise based way,
- Grown product and management structures to handle variability established in the last decade,
- Missing migration concepts for established and grown legacy systems to handle product variability,
- Increasing product and environment complexity,
- A constantly increasing number of variability drivers.

The only tool able to solve the problem is the formal logic (with different characteristics). Today, most of the variability tool vendors use operational logic that evaluates the rules in a particular order. This approach usually has unwanted side effects because different execution orders can lead to different results. As a consequence the encoding of variability in an operational logic is hard to debug. Furthermore, these tools are usually not complete which means that the tools do not consider all possible variants. These tools are already of big help, but will not succeed to solve the problem in an adequate way.

In order to obtain complete tools and avoid side effects, all rules have to be considered at once without any order. There are so called formal solvers that accomplish these requirements. However, the methods that these solvers implement are computationally expensive. This means that they are not feasible for industrial size problems, in general. As a consequence, the methods as well as their implementation in solvers have to be adapted to the respective input in order to obtain efficient solvers for dealing with the variability in industry. Although general purpose solvers can be applied to many industrial problems with some success, it is in our opinion not enough to be able to solve say 80 % of the problems and leave the rest with the so-called “local heroes”.

In addition, a solver alone does not solve the variability problems in industry because there is a strong need for a visualization that represents the solver results in an adequate way to the user. The visualization of product line models and their dependencies that represents the far end in product line engineering has to deal with huge variability models and often an incredible number of constraints (more than 200,000) for a complete product line. To the best knowledge of the authors there is currently no tool with an appropriate user interface available able to visualize such models, their constraints and analysis results from a solver in an effective and efficient way.

In order, to obtain such a comprehensive tool that efficiently analyzes product data and visualizes the models and the analysis results, requires two directions of research. One direction focuses on the development of reasoning procedures that efficiently analyze product data. The second direction of research focuses on the development of an appropriate user interface. With successful research in these areas, the product life cycle management would be revolutionized and there would be a significantly improving of product quality accompanied by a reduction of the development time and development costs.

This chapter presents current research towards this goal based on formal logic. Propositional logic has been recognized throughout the centuries as one of the corner stones of reasoning in philosophy and mathematics. With the help of proposition logic and sometime needed slight extensions to it, a wide range of combinatorial problems arising in variability management can be expressed and formulated as a propositional satisfiability (SAT) problem (see Sect. 17.3.2). Meanwhile SAT has become a reference problem for an enormous variety of complexity statements [24]. The presented method is based on SAT and its extensions that have been successfully used by the authors in several industrial projects. In addition, this chapter shows extensions of these methods and presents examples where these methods have been successfully applied. In addition, several directions for further research towards a comprehensive tool suit are sketched.

17.3 Functionalities for Management of Complex Products

In the concurrent engineering process, it is vital to have comprehensive analysis tools that ensure properties like consistency, correctness and ensure that specified properties are fulfilled. This is required throughout the whole life-cycle of the products. In particular, complex products can be composed of several thousand parts. In this context, complete analysis tools are indispensable that help the engineers to manage the product data, significantly improve product quality and reduce overall cost.

These tools identify errors in the development process as soon as they occur and can, therefore, be identified and avoided from the very beginning. This saves the possibly extremely expensive correction of these errors in a later stage of the development process or even after the product has been delivered to customers. The possibility to personalize a product in terms of individual customer preferences is nowadays one of the key factors in many industries.

Not only the presentation of variety but also the effective and efficient management of variety come into play. To address the mentioned challenges, there is a need for methodologies to specify, analyze and manage the occurring variety efficiently. In this chapter, we concentrate on the correct and complete analysis of industrial-sized variant descriptions. This is the computationally most difficult part moving the handling of complexity from the user to the algorithm. So, it is a powerful tool for the user, in order to benefit from the complexity of the products with respect to variety, quality and efficiency.

Below, examples are presented of an analysis task that a respective tool should be able to perform in order to successfully and completely manage all variants of a product:

- Detect inconsistencies and conflicts in the product variability description
- Determine all parts of a specific product
- Determine all parts that are currently not used in any product
- Determine in which products a particular part is used
- Compute product configurations with predefined properties, i.e. buildability
- Computation of the needed (optimal) variability for a product family
- Optimization of a product family according to predefined measures such as cost or customer orientation
- Identify all reusable components
- Identify products that are not desired by the costumers.

With such a tool it is not necessary anymore to restrict the number of variants as much as possible in order to maintain them. Moreover, it allows a manufacturer to maintain many more variants of a product by reducing the effort to manage all of them.

A tool providing this functionality must be functionally correct, complete, fast and efficient. In particular, it has to consider all possible product variants during the analysis. Approximations are not suitable for this purpose and must be excluded in this phase, because they do not consider all possible variants, i.e. they omit products during the analysis of a product portfolio. Thus, they do not guarantee to find all relevant information or errors in the product data. We argue that an incomplete approach is not good enough and, furthermore, can mislead and direct wrong decisions.

Because of the fact that discrete products have a discrete structure, complete methods are computationally expensive. In contrast to a continuous structure, a discrete structure means that a product is composed of several individual parts. Each part is either contained in the product or it is not contained. In general, complete procedures for discrete structures are at least NP-hard. For the worst case, this means that all combinations of all parts of a product have to be considered during the analysis. For example, for a product portfolio with 3,000 parts we have, in general, $2^{3,000}$ different products that have to be considered during the analysis. The same holds in case of engineering changes, which must be considered too.

To obtain efficient and complete analysis procedures for complex products, the encoding of the product data and the reasoning procedures have to match to one another. The research in the field of automatic theorem proving of the last decade has improved the underlying algorithms in such a way that product data containing several million product parts can be analyzed efficiently in many cases. In the remaining of this section, we present a complete method for the analysis of product data based on a particular kind of logical reasoning procedure, based on SAT [24] that was successfully applied under industrial conditions.

Figure 17.3 depicts the general workflow of analyzing and optimizing products according to the presented method. First the product data has to be transformed into a logical model representing all the relevant aspects of the product data (see Sect. 17.3.1). In the second step, the logical reasoning procedure that analyses and optimizes the logical product model (see Sect. 17.3.2) is applied. The result shown to the user depends on the performed analysis. In nearly all cases the result has to be adapted to the expected reader to be understandable and, thus, usable.

17.3.1 Logical Product Model

Transforming the product data into a representation in logic (as depicted in Fig. 17.3) defines a unique semantics for the data. Thus, logical reasoning tools can precisely analyze the logical model. In order to obtain an efficient and useful analyzing framework that supports the concurrent engineering process, the logical product model has to fulfill the following three substantial requirements:

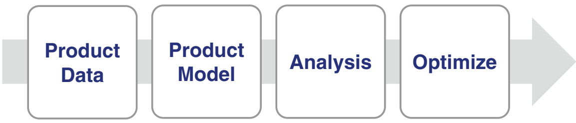


Fig. 17.3 Analysis process

- The model must adequately represent the relevant information of the original product data.
- The model must be in a form such that efficient analyzing procedures cope with model complexity.
- The structural representation of the model allows to trace back the results of the analysis to actual product characteristics.

Feature diagrams [25, 43] are an appropriate means, because they provide a logical framework that fulfills these requirements. They are suited to encode structural information of a product as well as the formulation of additional constraints.

This is an important step in order to build a knowledge base for all products of a manufacturer. In particular, this knowledge base is a tool that enables the engineers to consolidate their knowledge into one consistent knowledge base. The reasoning procedures shown in Sect. 17.3.2 provide a tool that automatically verifies the consistency of structural information and additional constraints, and prevents possible interface problems between several engineering departments. As a consequence, this knowledge base is an explicit representation of the manufacturer's know-how and enables everybody involved in the product lifecycle to access this knowledge (see Sect. 17.4.1).

Besides the mentioned advantages it should be noted that the creation of an appropriate logical representation of the product model is usually not a trivial task. It needs a lot of experience in the field of formal logic and their respective analyzing procedures. Furthermore, a gap remains between the real world product and the specified formal model. This gap can only be bridged with the help of the domain experts who should provide a crucial support within this first phase.

17.3.1.1 Feature Diagrams

Figure 17.4 depicts a feature diagram that represents an extension of the product structure given in Fig. 17.2. A feature diagram is a tree that represents a set of valid products which are also called variants. Thus, variants represent a valid selection of features (nodes in the feature tree) in a feature tree. It is possible to further restrict valid feature selections by so-called cross-tree constraints.

- Optional/mandatory features
  In every valid product a mandatory features (indicated by a filled circle in Fig. 17.4) is selected if and only if its parent node is selected. If an optional feature is selected, its parent has to be selected, too. The root of the feature diagram is usually mandatory and assumed to be part of all valid products.
- XOR, OR, AND group
  If a non-leaf feature F is set to XOR, exactly one of its children has to be selected if F itself is selected. In Fig. 17.4 the feature Engine is a XOR group, i.e. the car can have exactly one engine and in this case must have exactly one

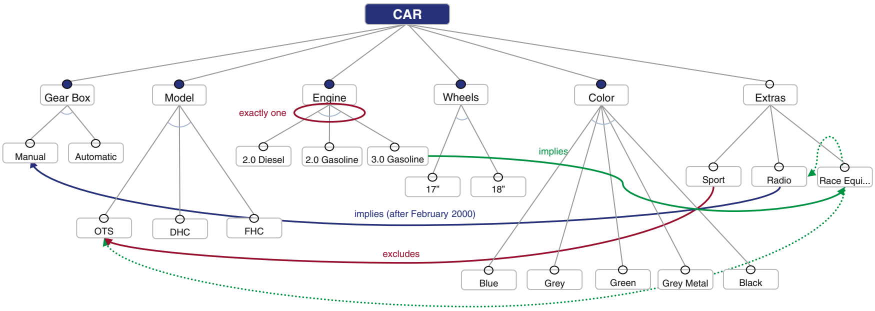


Fig. 17.4 Feature diagram

engine. Likewise, OR denotes that at most one children is selected and AND that all children have to be selected.

- Implies/Excludes
  The implies and excludes relations define so-called cross-tree constraints. For example, if the feature "3.0 gasoline" is selected then the feature "Race" has to be selected and if the feature "Radio" is selected then the feature open twoseater (OTS) is excluded, i.e. it cannot be selected.

Additionally, a feature diagram can contain arbitrary Boolean formulas over the features in the tree. The following formulas are examples for such constraints:

$$\text{Race} \rightarrow \text{Sport} \bigwedge \text{OTS}$$

This means whenever the Race option is part of a product then also the Sport option and the option OTS have to be included in the product.

The second example expresses the constraint that the OTS excludes a radio:

$$\text{OTS} \rightarrow \neg\text{Radio}$$

Feature diagrams represent product data with a precise semantics. Consequently, a feature diagram defines a mathematical model for the variability of the corresponding product data. This model constitutes the basis for the formal analysis described in Sect. 17.3.4.

17.3.1.2 Quantitative Extensions of Feature Diagrams

An extension of feature trees by attributes as mentioned by Benavides [26] extends the expressivity of feature models (see Fig. 17.5). Attributes provide a method to model additional information within the feature model. With appropriate reasoning methods these attributes can be used in the analysis process and the visualization. Examples for such attributes are costs, weight, and speed. There exist reasoning procedures for analyzing attributed feature trees and for optimizing in terms of the specified attributes.

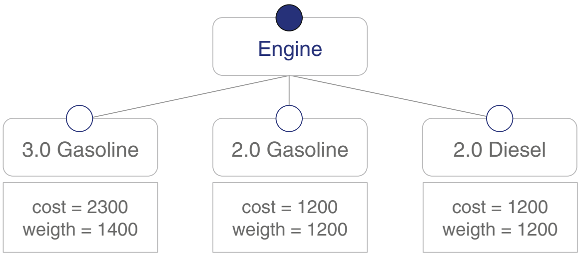


Fig. 17.5 Extended feature diagram

17.3.2 Logical Analysis

In order to perform logical operations on feature diagrams, the feature diagrams are first translated into a set of Boolean formulas. There exist tools that automatically verify the satisfiability of a set of Boolean formulas. The tools we consider here are so called SAT solvers [24].

All aforementioned analysis tasks can be formulated as a Boolean satisfiability problem and, consequently, can be answered by a SAT solver.

Aside from efficient SAT procedures, the right encoding of the real-world problem in Boolean logic, is the key for successfully analyzing industrial product data.

17.3.2.1 Translation into Logic

Feature diagrams are a formal representation of product data with a precise semantics. Consequently, they define a unique mathematical model of the product data that consists of the hierarchical structure and additional logical formulas that specify the properties and links to all product parts. The analyzing procedures presented here operate on purely logical representations of the product data. As a consequence, the structure and content of a feature diagram have to be transformed into an equivalent representation in logic.

The following example shows the translation of the node GearBox of Fig. 17.4 in propositional logic:

$$\begin{aligned} \text{GearBox} &\rightarrow \text{Manual} \vee \text{ManualCR} \\ \text{GearBox} &\rightarrow \neg\text{Manual} \vee \neg\text{ManualCR} \\ \text{Manual} &\rightarrow \text{GearBox} \\ \text{ManualCR} &\rightarrow \text{GearBox} \end{aligned}$$

This expresses the parent-child relation and the property that this node is an XOR node.

Although the translation of a feature model into a set of Boolean formulas is straightforward as proposed by Benavides et al. [26], this is a crucial step for successfully analyzing huge product data. Because they are computational expensive, the reasoning procedures rely on the right encoding of the input data in order to perform efficiently and to avoid the worst-case behavior.

From our experience, a respective encoding for most problems from industry can be found, which has also been proposed by Mendonca et al. [21]. However, this is not always obvious and might involve deeper inspection of the product data performed by the variability expert together with the domain expert.

17.3.2.2 Satisfiability Procedures for Feature Diagrams

The analysis tasks of Sect. 17.3.1.1 needs be transformed into a satisfiability problem. A satisfiability problem verifies for a set of Boolean formulas if there exists a variable assignment that fulfills all formulas at the same time.

Considering the following set of Boolean formulas where a, b and c are Boolean variables, i.e., they can be assigned true or false:

$$\begin{array}{ll} a \vee b \vee c & \neg a \vee b \vee \neg c \\ a \vee \neg b \vee c & a \vee \neg b \vee \neg c \\ a \vee b \vee \neg c & \neg a \vee b \vee c \\ \neg a \vee \neg b \vee c & \end{array}$$

Although this is a small problem, the solution is not obvious. If we consider problems with several thousand or million variables, this is almost impossible to do without efficient and sophisticated algorithms. In order to find a solution, one has to consider all possible assignments for the variables. Assuming a problem with 50,000 variables, we have to consider $2^{50,000}$ assignments. Once a fulfilling solution has been found, the solution is easy to verify. These kinds of problems are called NP-hard problems.

In 1960 an algorithm [27] was presented that computes an assignment for a set of Boolean functions. This algorithm was later improved and became known as the DPLL algorithm. The algorithm searches the decision tree as depicted in Fig. 17.6 for a satisfying solution.

In order to perform the satisfiability procedure efficiently, the DPLL algorithm has been improved by many techniques and methods. Tools that implement a satisfiability procedure are called SAT Solvers which are described in detail in the Handbook of Satisfiability [24]. Current SAT Solvers are mostly based on the

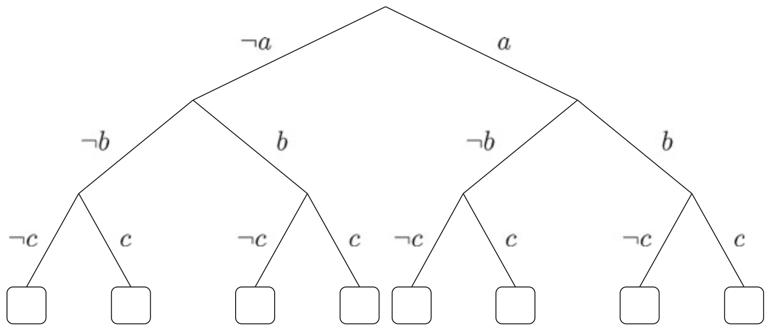


Fig. 17.6 Decision tree

Unit Propagation	$(M, N \cup \{C \vee L\}) \Rightarrow (M + L, N)$	$M \not\models C$ $L \notin M$ and $\neg L \notin M$	(1)
Decide	$(M, N \cup \{C \vee L\}) \Rightarrow (M + L^d, N)$	L occurs in N $L \notin M$ and $\neg L \notin M$	(2)
Fail	$(M, N \cup \{C\}) \Rightarrow \text{fail}$	M does not contain any decision literal $L \notin M$ and $\neg L \notin M$	(3)
Backjump	$(M' + L^d + M'', N) \Rightarrow (M' + L', N)$	$N \models C \vee L'$ $M' \not\models C$ $L' \notin M$ and $\neg L' \notin M$	(4)

Fig. 17.7 Abstract CDCL procedure

CDCL method that is an improvement of DPLL. The abstract DPLL calculus [44] is depicted in Fig. 17.7.

However, in general, it is not sufficient to use an off-the-shelf solver, because also the solver engine has to be adapted to the input problem in order to obtain the best, or in many cases an acceptable, performance. The harder the problems become, the more crucial it is to adapt the solver in order to solve the problem at all.

17.3.2.3 Reasoning Procedures for Extended Feature Diagrams

A feature diagram as well as an extended feature diagram has an equivalent representation as a set of logical formulas (see Sect. 17.3.2.1). The translation of an extended feature tree results in a propositional logic with an additional theory. For this kind of logic there exist three approaches of a reasoning procedure in basic research: modular [28–30], hierarchical [31, 32] and integrated approaches [33, 34].

The modular approach implements a black-box strategy whereas the other two approaches implement a white-box strategy:

(i) Modular approach: The modular approach adds the quantitative aspects as labels to the propositional formula.

$$\{\text{weight} = 5, \text{torque} = 450\}: \text{ManualCR}$$

(ii) Integrated approach: The integrated approach combines the quantitative aspects with the propositional formulas resulting in one formalism.

$$\text{ManualCR.data} = (5, 450)$$

(iii) Hierarchical approach: The hierarchical approach combines the quantitative aspects with the propositional formulas, too. In contrast to the integrated approach, the combination of these two formalism is only through variables.

$$x = 5 \wedge y = 450 \rightarrow \text{ManualCR.data} = (x, y)$$

The modular approach results in two almost independent formalisms. This means for the analysis that the theory part is not integrated in the actual analysis. Rather, it is analyzed independently with special methods for the respective theory. As a consequence, this method is called black-box strategy because the analysis procedure does not know anything about the theory methods. It only asks the theory black-box if a given label is valid in the theory part of the formalism.

The two white-box approaches combine the propositional and the theory part within a single formalism. On the one hand, this has the disadvantage that the two parts cannot be treated independently from each other. This results in methods that are considerably more complex. On the other hand, the white-box approaches are transparent. This means that the properties of the theory part of the formalism can be easier used in the propositional part in order to simplify the current formulas.

Adding taxonomies to the logical formalism of complex products is a further extension that requires decision procedures for the Bernays–Schönfinkel class. This is also called the effective propositional (EPR) class. A decision procedure for the Bernays–Schönfinkel class based on the first-order prover SPASS [35] is presented in [36]. An overview over EPR solvers can be found in [37].

It requires further investigation which of these approaches is best suited for analyzing the respective properties of industrial products.

17.3.2.4 SAT Based Optimization

In addition to the analysis, there exist efficient optimization procedures for product models that compute optimal products with respect to a given cost function. Because of the fact that product data have a discrete structure, computing an optimum requires discrete optimization procedures. In general, discrete optimization problems are much harder than continuous optimization problems. For further details about optimization in general and further use cases, Chap. 15 may be considered.

The hardness of discrete optimization problems for huge products such as cars, aircrafts or ships involving several thousand parts, requires highly efficient optimization techniques.

We have successfully used SAT based optimization procedures in several industrial projects applying the branch and bound method [38]. Using this procedure together with the logical product model (see Sect. 17.5.1), a defined cost model and an objective function computes optimum products with respect to various cost metrics.

For example, the following use cases can be solved with such an optimization procedure:

- Assembly sequence optimization [6],
- Computation of the optimal configuration, for example the cheapest, fastest, lightest product (see Sect. 17.4.2).
- Test coverage optimization.

An extended feature model as depicted in Fig. 17.5 defines a product model and a cost model. In addition to the objective function, these methods allow the formulation of additional bounds.

17.3.3 Visualization

In addition, to visualize the product structure as depicted in Sect. 17.3.1, the analysis and optimization results need an adequate visualization. The visualization should provide the results to the target user group or role, for example engineers, managers and sales people.

This means that the visualization of the same analysis task has to be visualized with respect to the specific user scope, considering specific needs or objectives e.g. level of detail or abstraction. The reasoning procedures shown in Sect. 17.3.2, produce the respective information.

However, preparing and visualizing this information such that the user efficiently can use it is an open area for research. Figure 17.8 illustrates an example for a reporting of a conflict. In this example, a user has selected two different kinds of wheels, but only one is allowed by the constraints. In the case that a conflict is more complicated and involves hundreds of features and constraints, this representation is not appropriate anymore in order to efficiently analyze the conflict.

Type	Constraint	Comment
User Selection	_Spoke_Wheel	User selection
Constraint	(imp(_Wheels exo(_Disk_Wheel _Spoke_Wheel)))	
FeatureTree	_Wheels	_Wheels is mandatory
User Selection	_Disk_Wheel	User selection
FeatureTree	_ExampleCars	_ExampleCars is mandatory

Fig. 17.8 Conflict reason

17.3.4 A Tool for Formal Variability Management

The presented analysis methods and the presented examples described in Sect. 17.4 are realized with the help of a commercial tool [39]. As opposed to tools as for example described in [40, 41], this tool focuses on the analysis and optimization of variant product structures based on the aforementioned propositional logic and a very efficient implementation of a SAT solving procedure. It offers a set of analysis possibilities as there are for example:

- Consistency Check
- Dead Feature Detection
- Optimization with bounds
- Product Configuration
- Analysis Result Visualization
- Formal Rule-based Debugging.

Considering these analysis procedures the focus of this tool, lies in the performance of the result computation. The user is now able to solve the specified problems in real-time. The tool allows for a systematic and timesaving analysis of huge product structures, keeping at the same time the right level of abstraction for the result presentation. It can be looked at as a formal logic based integrated development environment (IDE) for variant product structures where all the analysis results are mathematically proven to be correct. There is no longer a “perhaps”.

17.4 Applications

This section presents examples showing how SAT based methods can be used in order to significantly improve the overall quality of complex product data and, consequently, the quality of the actual products. As a consequence, they enable manufactures to offer more variants that they can manage. This allows them to satisfy the customer’s requirements with higher granularity.

The examples shown in this section are a product consistency check and product optimization.

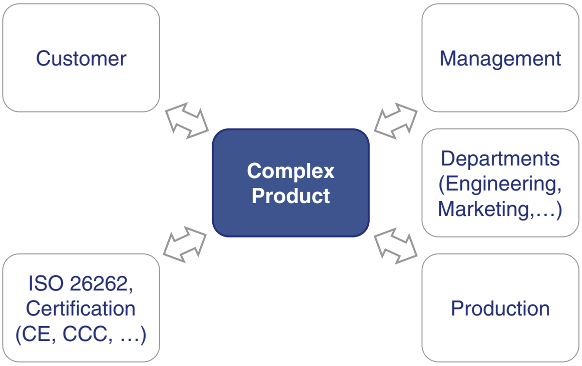


Fig. 17.9 Parties involved in the product lifecycle

17.4.1 Product Consistency

A complex product involves many parties, such as customers, mechanical engineering, electronic engineering, marketing, management, regulations and certifications. Ensuring that a product respects all requirements from all parties is a challenge that causes a lot of problems, in general. These problems are the major issue for product callbacks or severe damage to people and the manufacturers. Figure 17.9 illustrates the different parties that have particular requirements to a product.

Verifying that a product with several million variants satisfies all the requirements, is only possible with complete methods. Completeness means that all possible variants are respected and checked that they fulfill all specified requirements and satisfies all interfaces to all involved parties. This procedure is called a consistency check.

First, in order to use logical methods, the product model as well as the specifications must have been formally specified. Mostly, products and its specification are stored in a product data management (PDM) system. Consequently, these data have to be translated into a formal representation first.

In order to perform a consistency check for a product, all relevant aspects of the product that are not contained in the PDM system, have to be also present in a format that can be transferred to a formal model (see Sect. 17.3.1). For the estimation of costs, this could also involve enterprise resource planning (ERP) systems. Currently, there exists no general approach. Therefore, it is defined for every use case. This leaves room for research in order to find a general approach.

Even more, such a model improves the communication between the parties by defining a precise language that can automatically be verified against specified properties.

Once all the aspects are defined and transferred to a formal model, the formal methods of Sect. 17.3.2.2 verify their consistency. If there are inconsistencies, these methods will find them and generate an explanation. This explanation is called a proof. A proof is a mathematical precise representation of the reason of the inconsistency. Based on this proof, sophisticated and exact reports for each party involved can be generated together with proposals to solve them. This is an area for further applied research, too.

This section depicts the opportunities that formal methods provide in order to support the product lifecycle and to improve the product quality by ensuring that all requirements of all parties are fulfilled. At the same time, this section shows the areas for further research towards a general approach.

17.4.1.1 Industrial Application

The product model represents the information backbone for all people working on a specific product. In automotive industry for example, engineers must ensure that new parts fit into a well-defined set of car configurations—geometrically and logically. During the development phase the product structure and the geometrical parts are developed concurrently, thus both models are subject to a continuous change. In order to cope with the described situation, engineers work in so-called “reference configurations”. These reference configurations have to provide a valid reference for all product configurations using the designed parts, functions etc. Changes in the variant product model may have various effects on this: The reference configuration may represent the wrong set of configurations, or an incomplete set or the configuration becomes invalid at all. Obviously, managing complex products within a complex project environment with concurrently working engineers, this happens each and every day. The work of all mechanical designers, electrical engineers, software developers etc. deeply depends on a consistent product model. No person can detect all errors in the variant product model manually. The risk is that errors propagate through follow-up processes undetected, causing very large efforts and high costs. By this an automatic efficient consistency check of variant product models becomes an essential success factor for all companies offering variant products.

17.4.2 Product Optimization

The optimization methods in Sect. 17.3.2.4 allow the computation of optimum complex products with respect to specified metrics. Examples for cost metrics are price, weight and CO_2 emission.

The following shows examples for optimization tasks:

- What is the best product with respect to customer needs?
- What parts of a company portfolio are cheaper to be produced in house and what parts are cheaper to be bought from a supplier?
- What are the parts that cause the most costs?
- What parts are worth to consider for redesign in order to make them more cost efficient?

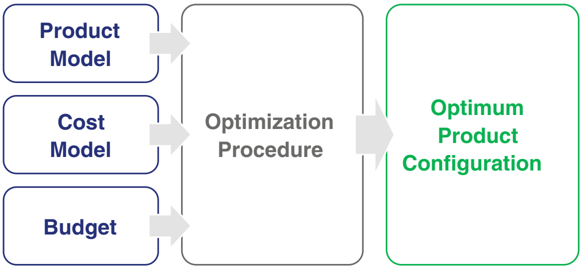


Fig. 17.10 Optimization of product configuration

Figure 17.10 depicts a black-box view to the optimization procedure for complex products. The inputs for the optimization procedure are a product model, a cost model and sales figures. The product model represents the product data and the cost model assigns certain cost values to the product model. The sales figure is a list of product variants that are expected to be sold or were sold in a defined time period.

For example, based on this input parameters the optimizer computes the cost-optimal product model by removing or adding parts and products to the product portfolio. The computed optimum is consistent in terms as explained in Sect. 17.4.1, i.e. all requirements are fulfilled. In addition to the consistency of the computed optimum, this method finds mathematically provable the global optimum.

The results of the optimization procedure can be the basis for further decision processes. They can optimize on several expected market situations and the respective consequences. In this case, the formal methods provide the presented functionality, but they have to be adapted to the individual application in order to perform efficiently. This is due to the fact, that an optimization operation is even more difficult from a computational point of view than a consistency check. Therefore, it is even more sensitive to the given product model. As a consequence, research towards a general approach for optimization of complex products is needed. Furthermore, there is currently no standard mechanism for the specification of the input parameters like the costs for the product model or the sales figures.

17.4.2.1 Industrial Application

Product variation is a key differentiator between competitors. However, product variation is causing costs in development, production and after sales too. The optimum is in-between the broadest offering to get more customers and the smallest offering to save costs. Real customer orders aren't evenly spread across the possible product configurations. It's the exact opposite: Real customer orders show accumulations of few sets of options, because there are always customer groups with similar requirements. One possible optimization is to offer packages with a well-defined set of options. These packages fulfill the requirements of many customers and reduce the number of variants and the costs too. However, the optimal set of packages is very hard to find, because no one can manually survey the mix of a complex cost structure, historical selling and forecast data within a highly variable product model. The identification of an optimal set of packages is the typical use case for a mathematical optimization toolkit. Even in this case the customer has to be taken more closely into account. Often the customer selects a package and afterwards wants to upgrade a certain feature within that respective package. Also in this case the optimization toolkit has to provide a useful variant.

17.5 Case Studies

In this section we present real industrial case studies for the application scenarios described in Sect. 17.4. These case studies were abstracted from the real case for obvious reasons.

17.5.1 Debugging a Product Variant Model

In this case study, a debugging tool for the product data of a globally operating machine manufacturer is presented. The debugging tool ensures that the product data fulfills the quality requirements. The manufacturer uses several mechanisms to express properties of his product involving engineering and marketing data. Altogether, the manufacturer defines several thousand of such product properties. At the engineering side the properties of the product describe all products that are buildable. The marketing side describes the market requirements and the product variants that are offered in the different countries.

The manufacturer uses a configuration tool that allowed the user to configure the product in the respective country. However, this process was error prone because the configuration tool did not offer a comprehensive analyzing mechanism, making the specification of the product model and the resulting variants transparent and understandable to the user.

The absence of such an analyzing mechanism results in a sometimes-unpredictable behavior of the configuration tool. Examining the specification mechanisms of the manufacturer in order to understand the reason for the behavior was very time consuming and required a lot of experience.

This has two reasons. First, the definition of the specification mechanisms was not unique and several exceptions to the rules were implemented. Second, the configuration engine used an operational semantics. This means the interpretation of the mechanisms was implemented in the software. As a result, the order of the rules had an impact on the behavior, which is usually an unwanted side effect.

In the case study, we used the commercial analyzing tool [39] that solves these problems by enabling und actively supporting the user to understand the problem

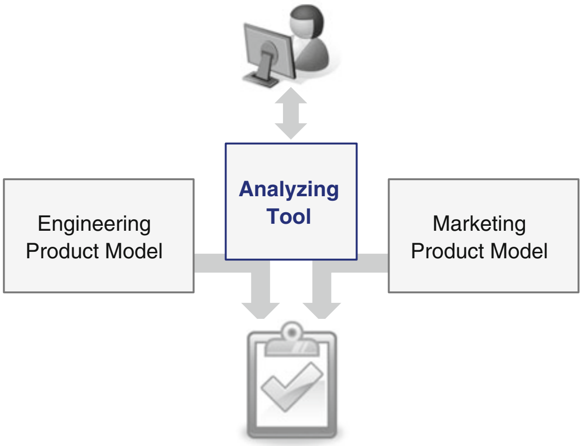


Fig. 17.11 Debugging process overview

and to find out whether the problem was a specification error or a bug in the configuration tool. Figure 17.11 depicts the debugging process and the respective inputs. For example, we discovered with the help of the tool that marketing offered a control unit for a variant of a machine for which the control software had not been released yet.

The first step in this investigation has been the translation of the specification mechanisms of the manufacturer into a logical product model. This defines a formal semantics based on mathematical logic for each individual mechanism and, as a consequence, each mechanism has a unique precisely specified meaning.

The logical product model can be loaded into the analyzing tool. The tool allows the user to interactively inspect the product model. Furthermore, the consequences of a selection of a feature in the product are computed in real time. Even more important, the user sees the reason for the consequences. In particular, the user immediately understands why the selection of one feature requires the selection/de-selection of other features.

In addition, the tool finds inconsistencies immediately and, therefore, avoids working with an incorrect product model. Usually a conflict involves just a few rules. As a consequence, the rules to be considered in order to understand the reason of an inconsistency is drastically reduced from several thousand to just a handful of rules.

Additionally, the tool enables even less experienced people to understand the specified product properties and its consequences. This is because the tool serves as a reference interpreter for the properties. As a consequence, the communication between several people involved in the product lifecycle of the product was significantly improved.

17.5.2 Computation of a Set of Cost-Optimized Wire Harnesses

In this case study the goal was to compute a set of cost optimum wire harnesses for a product portfolio of a manufacturer with respect to a forecast. The manufacturer wants to know which wire harness to order from a supplier in order to put in stock.

This is a crucial operation because the ordered harnesses have to fulfill the following requirements:

- For each product variant that is expected to be ordered there must be a matching wire harness in stock
- The wire harnesses in stock must be cost optimal.

This ranges from an individual wire harness for each order to one wire harness for all orders. The computation is based on the following inputs:

- The product portfolio and its properties describing the properties of the product variants and the mutual dependencies of its parts. This is defined by the manufacturer.
- Properties of the wire harnesses describing the dependencies between options in the wire harness. This is defined by the supplier.
- The costs of a wire harness representing the contract with the supplier. The contract defines all involved costs like production cost, stock costs, transportation costs and price deductions.
- An estimation of the expected sales for each product configuration relevant to the wire harnesses.

The current decision process of the manufacturer, which wire harness shall be ordered, is based on a local optimization method. Because of the fact, that the task is a discrete optimization problem, a local optimization can differ significantly from the global optimum (see Chap. 15). This requires a further step for additional optimization.

To compute the global optimum solution, we used the optimization engine of the commercial tool [39]. The tool guarantees that it finds an optimal solution. In order to use this tool we first translated the aforementioned input into a formal model with a defined set of cost attributes. This resulted in the following inputs for the tool, as depicted in Fig. 17.12:

- a model of the products,
- a model of the wire harnesses,
- a model of the costs of the wire harnesses and
- an estimate of the expected sales for each product configuration relevant to the wire harnesses.

Based on the logical representation of the products, the wire harnesses, the costs and the expected sales, the optimization engine computed a set of cost-optimized wire harnesses, i.e. the optimality can be proven mathematically. Because of the

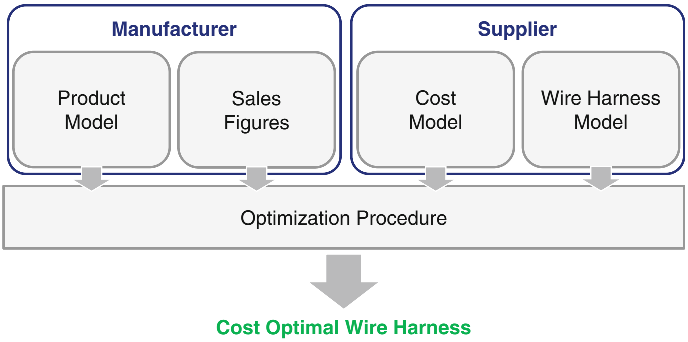


Fig. 17.12 Optimum product model

fact that a discrete optimization is computational expensive, the definition of the formal model is essential, because it has to fulfill two fundamental requirements. First, it must adequately represent the input data. Second, it must be defined in such a way that the optimization procedure works efficiently on the model.

The translation of the input data into a logical model resulted in 300,000 logical formulas. The optimization procedure took about 3 h to compute the global optimal solution. Comparing the optimization results of the tool with the current used method, results in an improvement of at least 15 %.

17.5.3 Car Pool Optimization for Manufacturers and Customers

In the case study car pool optimization, we examined the possibility to compute a cost optimum set of cars for the car pool of a company. The input parameters for this case study were the product model, a cost model for the product model, a given budget and a set of predefined options that every car should have. Then the optimization was on the fuel consumption of the whole car pool.

With current car configurators only one car can be configured with respect to the available options. The configurator computes the price and indicates the fuel consumption of the current configuration. In particular, there is no possibility to select a set of options and the remaining options are set with respect to an optimization procedure.

Further applications are the possibility to define preferences for a configuration and the optimization procedure finds the best car for a given budget. Different customers have their preferences for different kinds of options. For example, safety option, sport options or options causing the least fuel consumption.

Performing optimizations on configurations is a big advantage for both customer and manufacturer. The customer finds the product he wants quickly without bothering about options he is not interested in. The manufacturer on the other hand

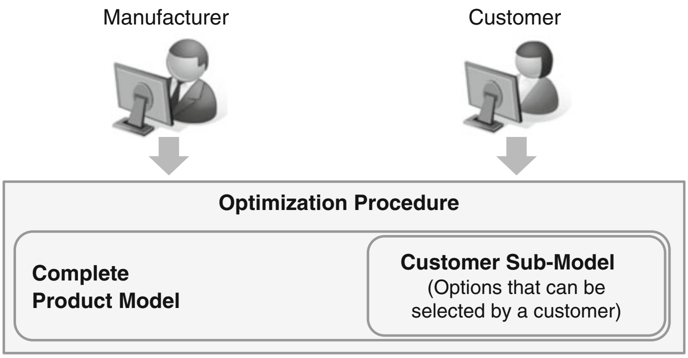


Fig. 17.13 Manufacturer and customer optimize on the same product model

can optimize his product portfolio particularly to the demands of the customers. The optimization enables him to better understand his product portfolio. These insights can be the basis for further decisions, for example launching of marketing offers or invention of new feature options.

Figure 17.13 depicts this case study. The manufacturer and the customer compute optimal configurations based on the same product model. For the customer, a subset of the available configurations is defined. For example, this defines the market offers of the manufacturer. Instead of having different representations of the product portfolio, we show in this case study that manufactures as well as customers can perform their processes on the same product model.

Furthermore, this method provides a marketing driven model based design and development of new products. In this case, the manufacturer starts with a less detailed customer model that defines the customer demands. The model becomes more detailed during the design and development phase of the product. Consequently, the current development model can be optimized with respect to the customer model and verified against the customer model.

17.6 Future Directives

In the late 90s, entering of electronics into products and customer demands for individual products to mass customer market prices caused a rise of the product complexity as indicated in Fig. 17.14. Electronics were shipped with diverse releases and software versions that have to exactly match the hardware parts. This has resulted in a huge number of variants that have to be maintained and have to fulfill specific properties. Further reasons for an explosion of variants are fast changing market demands, legal regulations, and certificates. Nowadays, the complexity of global products drastically rises due to the aforementioned reasons for variety explosion. On the other hand, the general purpose CDCL algorithm

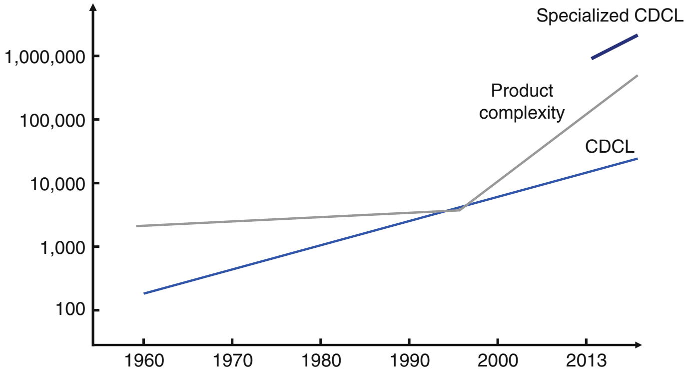


Fig. 17.14 Development of product complexity and the performance of general purpose CDCL procedures and specialized CDCL procedures

(see Sect. 17.3.2.2) has steadily improved such that its implementations could successfully analyze complex products. To be still able to successfully analyze these products with mathematical tools, the CDCL based algorithms can be specialized to the properties of these analyzing problems. An appropriate tool that implements a specialized CDCL algorithm and successfully analyses complex products is available [39].

The case studies in this chapter have shown that algorithms based on the methods depicted in this chapter are able to solve variant problems from various states in the product lifecycle that could not be solved without these. As a consequence, powerful analyzing tools exist that completely and precisely verify complex product models with respect to a diversity of properties involving variants. Furthermore, we expect that models are able to express variant information from other states in the product life cycle and the relations between the states as shown in Fig. 17.9. Examples of other states in the product life cycle are: the requirement and market analysis, design, construction, production, logistics, sales and after-sales.

In the area of model based systems engineering (MBSE) there is a standard modeling language, the systems modeling language (SysML) [42], with a diversity of analyzing tools for several aspects of the systems. In the case of variability management we expect a similar development towards a model based variability management (MBVM). The common variability language (CVL) [14] is an example of the development towards this direction. In the long term this could even go further towards a model based product life cycle management (MB PLM). From such a development we expect a similar impact on the development of products like crash simulations for the development of safe cars.

17.7 Conclusions and Outlook

This chapter has illustrated the crucial importance of variability management. We have presented some of the basic theoretical background and several real-world use cases that have been successfully mastered using methods based on formal logic. Thus, we have shown that the application of a formal logic-based approach to real-world variability problem scenarios is very promising. Although we had to customize the underlying analysis algorithms and find an appropriate way to encode the problem scenario in a formal setting, we are very confident to be able to find such optimizations for most problems of this kind. The more real-world problems we analyze, the more experience we gain in optimizing and customizing our algorithms and problem representations. Although we have to respect and observe the basic research that has been done and that will be done in the area, we are currently at a stage where we need more applied research based on real world examples in real industrial project settings to develop and improve a comprehensive tool suite based on formal logic. This will be one of the key factors enabling manufacturers and product line engineers to develop and master even more complex products and at the same time meet exactly the customer demands. To the best of our knowledge and having analyzed the past, variability reduction certainly is not an option for successful manufacturers or product line engineers in the future in a global competition. The only way for a sustainable variability management and product line engineering is to analyze, optimize, develop and adapt product variability in accordance with the rapidly changing market requirements. Formal approaches as described in this chapter provide the means for achieving this goal.

References

1. Hüttenrauch M, Baum T (2008) Effiziente Vielfalt Die dritte Revolution in der Automobilindustrie. Springer, Berlin
2. Elgh F (2014) Automated engineer-to-order systems. A task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3–4):324–347
3. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manag 7(2):101–115
4. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manag 7(2):116–131
5. Chang D, Chen CH (2014) Understanding the influence of customers on product innovation Int J Agile Syst Manag 7(3–4):348–364
6. Hu S, Ko J, Weyand L, ElMaraghy H, Lien T, Koren Y, Bley H, Chryssolouris G, Nasr N, Shpitalni M (2011) Assembly system design and operations for product variety. CIRP Ann Manuf Technol 60(2):715–733
7. Wallis R, Stjepandić J, Rulhoff S, Stromberger F, Deuse J (2014) Intelligent utilization of digital manufacturing data in modern product emergence processes. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, 2014, pp 261–270

8. Kretschmer R, Rulhoff S, Stjepandić J (2014) Design for assembly in series production by using data mining methods. In: Cha J et al. (eds.) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, 2014, pp 379–388
9. ElMaraghy H, Schuh G, ElMaraghy W, Piller F, Schönsleben P, Tseng M, Bernard A (2013) Product variety management. CIRP Ann Manuf Technol 62(2):629–652
10. ElMaraghy H, Azab A, Schuh G, Pulz C (2009) Managing variations in products, processes and manufacturing systems. CIRP Ann Manuf Technol 58(1):441–446
11. ElMaraghy W, ElMaraghy H, Tomiyama T, Monostori L (2012) Complexity in engineering design and manufacturing. CIRP Ann Manuf Technol 61(2):793–814
12. Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS (1990) Feature-oriented domain analysis (foda) feasibility study. Technical report, DTIC Document
13. Pohl K, Böckle G, van der Linden F (2005) Software product line engineering—foundations, principles, and techniques. Springer, Berlin
14. Haugen Ø, Wasowski A, Czarnecki K (2013) Cvl: common variability language. In: Kishi T, Jarzabek S, Gnesi S (eds) Proceedings of the 17th international software product line conference. ACM, New York, p 277
15. Nestor D, O'Malley L, Quigley AJ, Sikora E, Thiel S (2007) Visualisation of variability in software product line engineering. In: Pohl K, Heymans P, Kang KC, Metzger A (eds) 1st International workshop on variability modelling of software-intensive systems (VAMOS 2007). http://www.sse.uni-due.de/vamos/2007/files/vamos07_0038_paper_7.pdf. Accessed 20 Aug 2014
16. Heuer A, Lauenroth K, Müller M, Scheele JN (2010) Towards effective visual modeling of complex software product lines. In: Botterweck G, Jarzabek S, Kishi T, Lee J, Livengood S (eds.) Software product lines: going beyond: Proceedings of the 14th international software product line conference, pp 229–238, SPLC 2010. Jeju Island, Sep 13–17 2010, Springer, Berlin
17. Botterweck G, Thiel S, Nestor D, bin Abid S, Cawley C (2008) Visual tool support for configuring and understanding software product lines. In: Geppert B, Pohl K (eds) Proceedings of the 12th international software product line conference, SPLC 2008. IEEE Computer Society, Los Alamitos, pp 77–86
18. Cawley C, Nestor D, Preußner A, Botterweck G, Thiel S (2008) Interactive visualisation to support product configuration in software product lines. In: Heymans P, Kang KC, Metzger A, Pohl K (eds.) 2nd international workshop on variability modelling of software-intensive systems (VAMOS 2008). http://www.sse.uni-due.de/vamos/2008/papers/VAMOS08_01.pdf. Accessed 20 Aug 2014
19. Schneeweiss D, Botterweck G (2010) Using flow maps to visualize product attributes during feature configuration. In: Botterweck G, Jarzabek S, Kishi T, Lee J, Livengood S (eds) Software product lines: going beyond. Proceedings of the 14th international software product line conference, pp 219–228, SPLC 2010. Jeju Island, 13–17 Sep 2010, Springer, Berlin
20. Czarnecki RK, Helsen S, Eisenecker UW (2004) Staged configuration using feature models. In: Nord RL (ed) Software product lines: 3rd international conference, SPLC 2004, 30 Aug–2 Sep 2004, Boston. vol 3154 LNCS, pp 266–283, Springer, Berlin. http://www.ece.uwaterloo.ca/~kczarnec/splc04.pdf. Accessed 20 Aug 2014
21. Mendonca M, Wasowski A, Czarnecki K (2009) SAT-based analysis of feature models is easy. In: Muthig D, McGregor JD (eds) 13th international software product line conference, SPLC 2009, 24–28 Aug 2009. San Francisco, ACM international conference proceeding series, 446:231–240, ACM. http://gsd.uwaterloo.ca:8088/SPLOT/articles/mendonca_sat_analysis_splc_2009.pdf. Accessed 20 Aug 2014
22. Antkiewicz M, Czarnecki K (2004) FeaturePlugin: feature modeling plug-in for eclipse. In: Burke G (ed.) Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange. pp 67–72, ACM. http://gp.uwaterloo.ca/sites/default/files/2004-antkiewicz-feature-modeling-plugin_0.pdf. Accessed 20 Aug 2014

23. Capilla R, Bosch J, Kang KC (2013) Systems and software variability management: concepts, tools and experiences. Springer, Berlin
24. Biere A, Heule M, van Maaren H, Walsh T (2009) Handbook of satisfiability. Frontiers in artificial intelligence and applications, vol. 185. IOS Press, Amsterdam
25. Kang KC, Kim S, Lee J, Kim K, Shin E, Huh M (1998) FORM: a feature-oriented reuse method with domain-specific reference architectures. Ann Softw Eng 5:143–168
26. Benavides D, Segura S, Ruiz-Cortes A (2010) Automated analysis of feature models 20 years later: a literature review. Inf Syst 35(6):615–636. http://www.researchgate.net/publication/223760542_Automated_analysis_of_feature_models_20_years_later_A_literature_review/links/0046352bd57ee8f1c9000000. Accessed 20 Aug 2014
27. Davis M, Putnam H (1960) A computing procedure for quantification theory. J ACM 7 (3):201–215
28. Nelson G, Oppen D (1979) Simplification by cooperating decision procedures. ACM Trans Program Lang Syst 1(2):245–257
29. Ganzinger H, Sofronie-Stokkermans V, Waldmann U (2006) Modular proof systems for partial functions with Evans equality. Inf Comput 204(10):1453–1492
30. Nieuwenhuis R, Oliveras A, Tinelli C (2006) Solving sat and sat modulo theories: From an abstract davis–putnam–logemann–loveland procedure to dpll(t). J ACM 53:937–977, Nov 2006. http://www.divms.uiowa.edu/ftp/tinelli/papers/NieOT-JACM-06.pdf. Accessed 20 Aug 2014
31. Ihlemann C, Jacobs S, Sofronie-Stokkermans V (2008) On local reasoning in verification. In: Ramakrishnan CR, Rehof J (eds) Tools and algorithms for the construction and analysis of systems. 14th international conference, TACAS 2008, held as part of the joint european conferences on theory and practice of software, pp 265–281, ETAPS 2008, Budapest, March 29–April 6 2008. LNCS vol 4963, Springer, Berlin
32. Althaus E, Kruglov E, Weidenbach C (2009) Superposition modulo linear arithmetic sup(la). In: Ghilardi S, Sebastiani R (eds.) Frontiers of combining systems, Proceedings of 7th international symposium, pp 84–99, FroCoS 2009, Trento, September 16–18, 2009. LNCS vol 5749, Springer, Berlin
33. Waldmann U (2001). Superposition and chaining for totally ordered divisible abelian groups (Extended abstract). In: Gore R, Leitsch A, Nipkow T (eds) Automated reasoning : first international joint conference, pp 226–241, IJCAR 2001, LNAI, vol 2083, Siena, 2001. Springer, Berlin
34. Korovin K, Voronkov A (2007) Integrating linear arithmetic into superposition calculus. In: Duparc J, Henzinger TA (eds.) 21st international workshop, CSL 2007, 16th annual conference of the EACSL, Lausanne, Switzerland, 11–15 Sept 2007, vol 4646 LNCS, pp 223–237. Springer, Berlin. http://pdf.aminer.org/000/563/602/a_precedence_based_total_ac_compatible_ordering.pdf. Accessed 20 Aug 2014
35. Weidenbach C, Dimova D, Fietzke A, Kumar R, Suda M, Wischnewski P (2009) SPASS Version 3.5. In: Schmidt R (ed) CADE—22, 22nd international conference on automated deduction, pp 140–145, Montreal, 2–7 Aug 2009, LNCS, vol 5663. Springer, Berlin
36. Suda M, Christoph W, Patrick W (2010) On the saturation of YAGO. IJCAR, pp 441–456
37. Sutcliffe Geoff, Suttner Christian B (2006) The state of CASC. AI Commun 19(1):35–48
38. Larrosa J, Nieuwenhuis R, Oliveras A, Rodriguez-Carbonell E (2009) Branch and bound for boolean optimization and the generation of optimality certificates. In: Kullmann O (ed.) Theory and applications of satisfiability testing—SAT 2009, 12th international conference pp 453–466, SAT 2009, Swansea, 30 Jun–3 Jul 2009, vol 5584 of LNC, Springer, Berlin. http://www.researchgate.net/publication/220944553_Branch_and_Bound_for_Boolean_Optimization_and_the_Generation_of_Optimality_Certificates. Accessed 20 Aug 2014
39. Logic4Business GmbH: www.logic4business.com
40. BigLever Software, Inc.: www.biglever.com
41. Pure systems GmbH. www.pure-systems.com
42. Object Management Group. http://www.omgsysml.org/

43. Bontemps Y, Heymans P, Schobbens PY, Trigaux JC (2004) Semantics of FODA feature diagrams. In: Männistö T, Bosch J (eds.) Proceedings SPLC 2004 workshop on software variability management for product derivation—towards tool support, pp 48–58. http://www.researchgate.net/publication/234080947_Semantics_of_FODA_feature_diagrams/links/00b4952691aadd1ab6000000. Accessed 20 Aug 2014
44. Nieuwenhuis R, Oliveras A, Tinelli C (2004) Abstract DPLL and Abstract DPLL Modulo Theories. LPAR 2004:36–50

Chapter 18
Intellectual Property Protection

Josip Stjepandić, Harald Liese and Amy J.C. Trappey

Abstract With the growth of the knowledge-based economy, intellectual property right (IPR) is recognized as a key factor to develop and protect strategic competitiveness and innovation of an enterprise. The increasing degree of collaboration in global relationships, ubiquitous digital communication techniques as well as tough competition has lead to an increasing importance of intellectual property protection (IPP) for enterprises. Since the law as well as ethical principles are not always adhered to, there are increasingly activities outside legal understanding. This situation is exacerbated in the context of rising crime through the misuse of modern ICT technologies ("Cyber Crime") and now employs extensively state authorities. Piracy, counterfeits and unwanted know-how drain pose a significant problem for each market leader. Intellectual property is stored in product data too. Especially modern parametric and feature-based 3D-CAD systems have been enhanced towards acquiring, representing, processing and distributing knowledge to support knowledge-based engineering (KBE) within virtual product creation. However, it is very easy to exchange huge amounts of product data within a virtual enterprise that comprises an enterprise with its supplier network. There is an enormous threat that intellectual property could fall into the wrong hands and badly jeopardize the existence of the related company. This chapter contains an analysis of this conflict area, a picture of the legal framework, a discussion on the need for action in supply chain networks and attempts by research and development as well as best practices in industry for various aspects of IPP in the context of concurrent engineering (CE).

J. Stjepandić (✉) · H. Liese
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: josip.stjepandic@prostep.com

H. Liese
e-mail: harald.liese@prostep.com

A.J.C. Trappey
Department of Industrial Engineering and Engineering Management,
National Tsing Hua University, Hsinchu 300, Taiwan
e-mail: trappey@ie.nthu.edu.tw

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_18

Keywords Product innovation · Virtual product creation · Knowledge-based engineering · KBE · Intellectual property protection · IPP · Patent infringement

18.1 Introduction

Continuous innovation and inventions are the basis for competitiveness and success of leading enterprises in a global context ("market leaders"). Fostering their excellent know-how they have managed in recent years to expand their position in the global market place. Therefore, they are able to reap the rewards of their extensive investments in research and development. Such enterprises accumulate also large amounts of intellectual property which is a broad, summarized label for the set of intangibles owned and legally protected by an enterprise from outside use or implementation without consent. In general, it consists of business know-how (product, process, service) and rights (patents, trade secrets, copyrights and trademarks) which can be more valuable than the tangible assets.

However, development and innovation of the technologies used to create these intangible assets need new approaches and technologies to legally maintain, protect and safeguard intellectual property. The concept of intellectual property implies that certain products of human intellect should be afforded the same protective measures that apply to physical property. Leading industrial nations have already developed a legal framework to protect both forms of property. In this era of globalization, threats are global too and all measures against loss and theft of IP must be considered globally in a new dimension.

In the context of concurrent engineering (CE), intellectual property protection (IPP) assumes particular importance. Global collaboration through virtual enterprise, high transparency of processes and data, high level of phasing in business processes, knowledge-based methods, intensive use of IT methods and tools, and high frequent data exchange are all features of a well implemented CE in each enterprise [1]. Hereby, however, the door is opened for all kinds of violations of intellectual property, if no protection is considered. Extracting value from intellectual property and preventing others from deriving value from it is an important responsibility for any enterprise. Therefore, this chapter will be focused on new methods and technologies and adds to the approaches introduced and explained in Chap. 10 [Knowledge-based engineering (KBE)].

The structure of this chapter reflects this focus. In the following Sect. 18.2, the political and socio-economical backgrounds for IPP are briefly introduced. It explains why IPP has become so important in the past years. The hot spot of IPP in engineering collaboration is highlighted in Sect. 18.3. In the following Sect. 18.4, the technical scope of this chapter is aligned with phases of the product creation process and main directions are drawn. Supporting methods for tracking of patent infringements are explained in Sect. 18.5. Subsequently, general methods for IPP protection in the information flow are derived in Sect. 18.6, followed with techniques for protection of product data and several showcases. For sake of completeness, the basic methods for protection of product and parts are inserted to Sect. 18.7. A conclusion section gives insight into future research and development of IPP from a CE perspective.

18.2 Political and Socio-economical Background

In the past decade, globalization was the dominant trend in the world economy. So called BRIC countries (Brazil, Russia, India, China) have become more and more important accompanied by a rising economic and political liberalization. Successful enterprises have anticipated this trend and got involved in those countries by their own subsidiaries and joint ventures which include all the functions of a global, decentralized enterprise. This induces the need to consider different political constraints. One of the most important constraints among others is the request by many governments to share and transfer know-how to joint-ventures and subsidiaries across the whole supply chain located in their own country. The intention behind that requests is to build-up self-sufficient, sustainable industries which are not dependent on know-how and technology transfers from abroad any more. While such build-up of competitors with military precision is not to interest of any investor, he will seek for appropriate measures to keep his competitive advantage in each world region [2].

Also, in every society a continuous transformation is going on by the influence of social media and mobile applications. In certain applications (e.g., social networks) it is increasingly difficult to distinguish between professional and private use. The impression for an average internet user is that all content for which access is not restricted is for free. Free file-sharing platforms are an example only. Furthermore, there are more and more social and political movements, to work towards the introduction of a “Digital Utopia” (free access to entire digital content). Unrestricted sharing of data has become a feature of many sub-cultures, regardless any statutory provision. An attempt of the US government to react radically by shutting down foreign sites that are accused of criminal copyright infringements, has caused the largest online protest in history [3].

Against this political and socio-economic background, where many stakeholders stimulate sharing and policing illegal internet activity remains difficult, it is not easy to distinguish between public and confidential as well as secret and confidential content [4]. In case of sensible product data which are exchanged across the globe this topic becomes hot. On the other side, in case of property rights the rising challenge is to identify their violation not only in the same country or region (e.g. European Union), but everywhere in the world. Thus, the thief could use the stolen content for production on a singular location far abroad, and compete with the legal owner for sales on the global market. This gives the protection of intellectual property right (IPR) a particular importance.

In an essay Azevedo et al. [5] investigate the relationship between protection of IPR, technological change and endogenous economic growth. "Despite the divergence of results regarding theoretical studies, most empirical studies find a net positive effect, which means that positive effects of IPR protection outweigh the negative effects. A possible explanation for this is that the empirical measure of patent protection, which is typically used, is just a summary of the statistics relating to the different categories of patent rights and so it is not clear how each type of patent rights influences innovation on empirical grounds" [5, p. 60].

Nevertheless, almost every week the newspapers publish news about various forms of intellectual property violation: product piracy, plagiarism, counterfeits and theft of crucial data. The victims of this new kind of international crimes are the global enterprises as well as small or medium enterprises. In many cases the victims reported really curious stories like the company Doppelmayr from Austria, which claimed a peculiar example of extremely professional product piracy and counterfeits [6]. A report on the patent law-suit between Apple and Samsung looks to the neutral observer like an intentional continuation of market competition in another, legal, field [7]. Finally, governmental bodies claim new kinds of crime ("cyber crime"), too [8]. In claim reports Chinese governmental bodies are accused of controlling and supporting cyber attacks against western companies and institutions [9]. According to a research study conducted during 2013, almost each enterprise in Germany was at least once the subject of a cyber attack [10].

Based on complaints from German machine tool manufacturers—according to them Chinese competitors violate their patent rights—researchers [11] have considered the situation of one century ago, when American manufacturers raised the same complaints against their German competitors. They came to the conclusion that the willingness of each market participant to respect IP rights is associated with its own competitive position.

Security experts view this problem in a similar way and point out at the early beginning of their book: "Illegal and furthermore criminal activity is a reality of the world in which we live… It is, however, a recurring theme, which bears repeating in all spheres of public live. One question we are often asked is whether there is any hope in combating this activity…. Moreover, it is not a trivial undertaking and should not be presented in a light that either under-emphasizes or over-aggrandizes it" [12, p. 3].

The amount and frequency of violations of IPR and their estimated losses suggest that currently the significant and crucial prerequisites for successful protection of IPR in many world regions are not fulfilled. Hence, the distinctiveness of the current situation lies not in the nature of IPR itself, but rather in the fact that political, socio-economic, organizational and technical constraints for an expected violation of IPR are particularly favorable [13]. This condition could persist because there is no international initiative whatsoever apparent that would revert this situation. Recent research predicts expansion of IPR by 2025, including enforcement [14, p. 358].

Based on that background all regulations concerning intellectual properties and the authentication of products and parts need to be expressed in a *protection policy* which on the one hand governs individual enforcement mechanisms implemented at the various sites and, on the other, integrates these protection issues into overall product lifecycle management (PLM) in an explicit way. These regulations should propose suitable measures that counteract violation of IPRs and infiltration of counterfeits. Since such threats are anticipated, ubiquitous protection measures have to cover the whole product lifecycle. This perception has now taken hold in the boards of almost all technology companies. Under high pressure corresponding concepts are currently developed or in preparation [15].

Further explanation in this chapter will be focused on technical aspects in the context of CE. For smaller companies which have no possibility to exclusively employ intellectual property experts innovative concepts and guidelines like guideline "Leitfaden zum Produkt- und Know-how-Schutz (Guideline to Product and Know-how Protection)" [16] are foreseen which is prepared by a special expert group of VDMA (Germany Association for Machinery and Process Industry). Based on various industrial scenarios the authors demonstrate their best practices and recommendations for dedicated use cases. Such concepts may support innovation by offering reuse of intellectual property knowledge and intellectual property-friendly design. Also risk monitoring via intellectual property asset tracking can be included.

In trying to relate IPR concepts and methods to the entire product lifecycle, it must be taken into account that, in general, patents are independent of any particular product and project. Therefore, patents must be maintained and defended as long as their subject has a commercial value for (potential) competitors. Product data in a project must be intensively protected from start of development (SoD) to start of production (SoP) and possibly less intensively afterwards (e.g., if they are created in a collaborative project that will be subsequently diminished). From the SoP until the end of a product life cycle (EoL), growing focus will be on the protection of products, spare parts and embedded software against possible counterfeits (Fig. 18.1). In that case reverse engineering (see Chap. 12) must be prevented or at least inhibited.

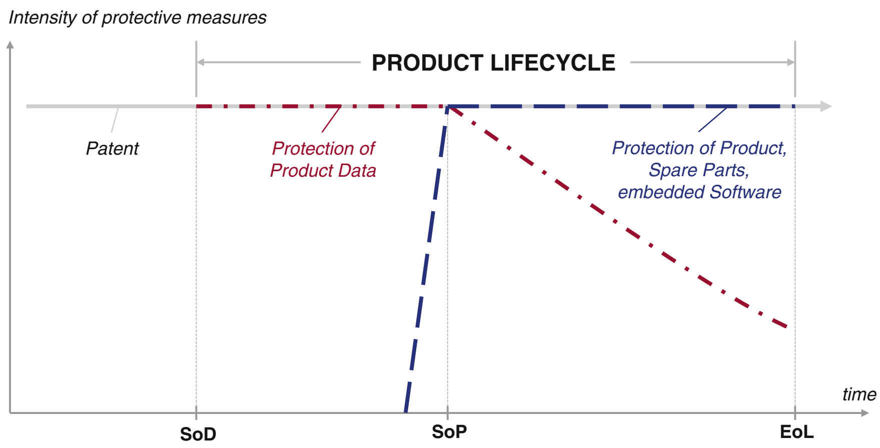


Fig. 18.1 IP protection in relation to product lifecycle

18.3 Hot Spot Engineering Collaboration

In many industries (automotive, aerospace, shipbuilding) product development usually takes place in global development partnerships. Original equipment manufacturers (OEM) conduct the development of new products at many locations in several countries across the world. Furthermore, a variable number of external service providers and suppliers take part in individual projects (see Chap. 7). Most relationships in a supply network which acts like a virtual enterprise are temporary, extinct by the end of project, while the contract can be renewed or expire (Fig. 18.2) [17]. In the latter case today's a project partner can become tomorrow's harshest competitor [1].

The IT infrastructure has to be adapted continuously to the moving constellation of project partners. Large supplier network implies a frequent mutual data exchange. Therefore, different solutions for global data logistics are available for the exchange of 3D data especially for the automotive industry [18]. Since data exchange runs on a regular, daily, basis with increasing amounts of data, the importance of know-how protection becomes evident to avoid knowledge leakage in such a complex, diverse network. For this purpose many OEMs allow the use of rich clients of their product data management (PDM) systems only at the supplier site. This monolithic solution provides secure access to the OEM's engineering database by standardized processes like authorization and authentication of singular users and devices. That simplifies the processes on the OEM side reducing a supplier to few authorized contact person who act manually as simple, independent users. On the other hand, such approach inhibits any supplier's IT integration with OEM (see Sect. 21.7) and imposes a permanent data queue on the supplier's side.

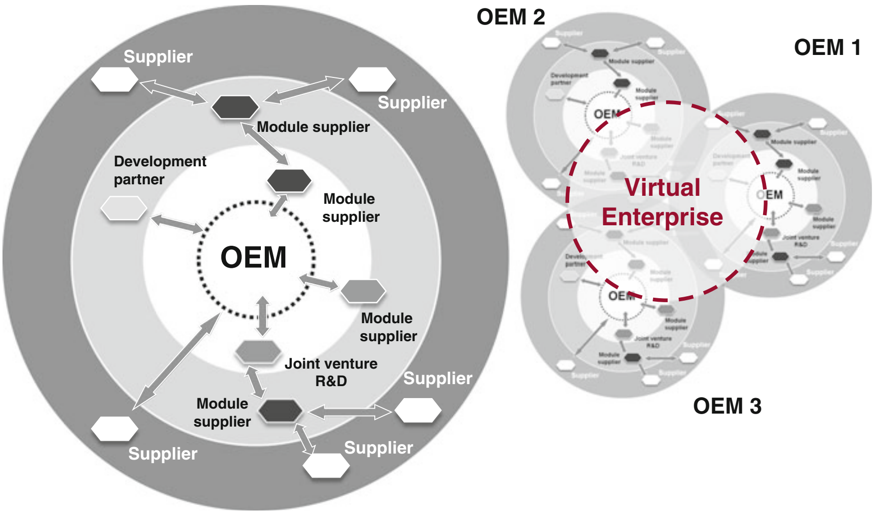


Fig. 18.2 Virtual enterprise in automotive industry

There are fundamentally opposing interests of the parties involved in the collaborative product development in a virtual enterprise. The customer (e.g., the OEM) normally commissions its suppliers and/or external services providers to provide components and/or explicit services [19]. So, in principle, the OEM is interested in the whole of technology and know-how. However, suppliers often complain that draft designs provided by them to the OEM were later given to their competitors during the concept competition phase. In the majority of cases suppliers wish to reuse developments achieved in one project in the next project with possibly another OEM [5]. The first OEM that acts as a client and financer of the new development may see the danger that he indirectly supports its competitors if his supplier sells the same technology to other OEM. This conflict of interests has existed before and has got additional explosiveness by the employment of KBE modules, because more and more valuable knowledge is integrated into computer-aided design (CAD) models and, as such, in daily data exchange. It is self-evident that such challenges can't be resolved by contractual regulations only, but require conceptual and organizational measures, and suitable software solutions.

The increasing number of plagiarisms, which invade certain markets, is a clear indicator for the intensive, systematic, but naturally not intended leakage of knowledge during a product development process [20]. Both manufacturing enterprises and independent research organizations are concerned with risks estimation regarding the loss of intellectual property [21]. The Aberdeen Group has surveyed 88 companies [22] to determine the degree to which best-in-class organizations are using security solutions to address the risk of so-called insider threat. Both supplier and service provider can be classified as insider, e.g., in the context of on-site design work or direct access to a partner's PDM system as aforementioned. The results uncovered that the majority of respondents have yet to implement technology to address insider threats. Only the best-in-class companies reported the decrease of security events, while in "laggard" companies security events significantly increased.

The damage is larger the earlier the product information is stolen and the later the piracy is discovered [21]. The situation is especially critical in the case of products with short lifecycles when product information (e.g., CAD data) is lost or stolen in an early phase of product development. There are known cases of plagiarism where products which are produced based on stolen product data, were introduced into the market earlier than the original product. In summary, there is a strong demand of both OEMs and suppliers for research on and development of methods for IP protection driven by their specific knowledge integration (see Sect. 10.5) and application of design methods [23]. Specific needs arise in long-lasting relationships where crucial key factors (type of collaboration content, degree of maturity, lot size) fundamentally evolve [24]. Meiwald has developed a generic approach for fundamental threat analysis and elaboration of basis protection concepts [25].

18.4 Technical Scope

Looking at Fig. 18.1, it can be recognized at first that patent management must follow technology management, i.e., patent management is strongly linked to the lifecycle of technologies starting with the discovery of ideas and continuing until a product is discarded from the firm's portfolio. Based on this, five distinctive phases reflect the patent life cycle management activities: explore, generate, protect, optimize, and decline [26]. While we are focused on protection of patents, Bryer et al. [27] give a comprehensive insight into the legal operations and implementation of IP. A patent is an IPR that is granted with exclusive right for commercialization and can potentially bring huge profit to an enterprise. Due to the fast growth of patents and infringement litigation, enterprises use a variety of patent strategies to protect their intellectual property or defend their IPR against malicious litigation. At first, patent infringement analysis shall be conducted frequently during the patent lifecycle. For example, a patent search and patent map analysis system supports engineers to find and analyze patents effectively. This is a field which obviously can be tremendously supported by appropriate IT methods and tools. However, patent infringement analysis still requires a large number of intellectual property engineers and lawyers to analyze patent claims and access the degree of infringement. Therefore, their work can be fostered by patent ontology engineering (POE) to enable an intellectual property defense support system (IPDSS). POE aims to analyze rapidly the structure of patent claims and their technical components [28].

Looking again at Fig. 18.1, product data is the next point of interest for protection. Nowadays, the majority of the manufacturing industry is deploying modern engineering IT technology and improving their usage by new application methods. Continuous optimization includes the deployment of technologies like CAD and PDM into virtual product creation and the application of new functionalities of further authoring systems. Techniques of artificial intelligence (AI) offer room for further improvements. AI covers many methods for acquiring, processing and storing of knowledge, which are systematically examined by Hopgood [29] (see Sect. 10.2). In the technologically leading 3D-CAD systems AI methods have been embedded, facilitating inclusion of product data (e.g., functional elements, computations, optimization goals and rules, whole templates) in a single data model (see Sect. 10.5). With such embedded KBE technologies interaction of comprehensive product-specific knowledge and know-how regarding the design process is made possible with a single CAD model [30]. Various validation approaches and procedures can be also directly embedded or deeply integrated by using external applications (e.g., Functional Digital Mock-Up, see Sect. 13.6) based on KBE technology.

In the context of PLM effective implementation of an integrated process chain is primarily based on the data, the information and the knowledge as represented in the 3D-CAD model. In CAD models different types of knowledge as well as representations of design methods can be stored (Fig. 18.3) [31]. PDM systems not only manage CAD models but also contain meta-information and workflow, while

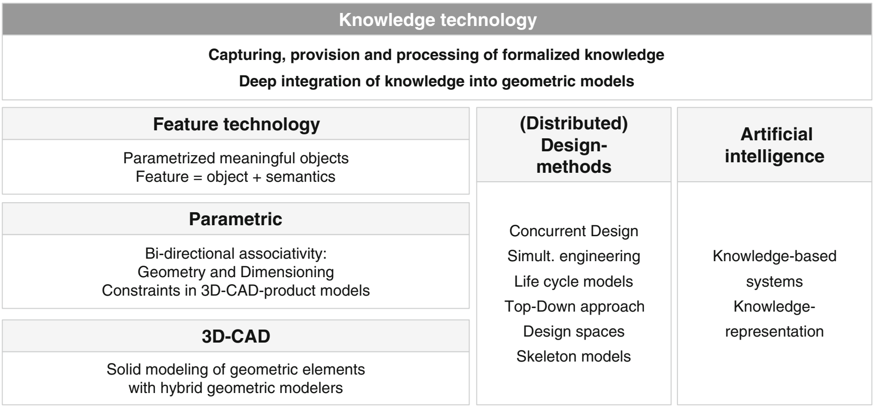


Fig. 18.3 Knowledge integration in 3D-CAD systems

they also document the entire product life cycle. Well customized, a PDM system can be considered as a storage container of formalized product know-how within an enterprise.

Product and process knowledge has to be managed and updated continuously in an enterprise [32]. Its usage has to be kept attractive for all potential users to provide appropriate design reuse at any phase of a product creation processes [33]. Ong et al. [34] depict a broad figure of methodical approaches to systematic design reuse from basic design, optimization, costs and performance evaluation to the definition of product families. Furthermore, the modular design implies the capture of knowledge needed for proper reuse of modules. Consequently, CAD models represent an important portion of the knowledge base of each product-centric enterprise, which after their generation must be stored, maintained, archived and protected during a whole product lifecycle (see Sect. 10.4). Figure 18.4 illustrates typical product knowledge items and design methods taken from customer requirements in the CAD system CATIA V5 that should be protected by IPP measures. For better classification the singular CATIA V5 items in Fig. 18.4 are placed in the general scheme explained in this section and Fig. 18.3.

The chances, which deep knowledge integration offers, are very promising and widely used to optimize a virtual product creation process. It is inevitable that this progress has also had a considerable impact on the way how to perform sophisticated development tasks. The achieved return on investment (ROI) is tremendous in many cases [31]. KBE templates became a significant accelerator for design projects by considerable savings in time and costs (see Sect. 10.6). For such reasons they are a real asset for each enterprise.

However, deep knowledge integration is accompanied by potential risks because each document can fall into the wrong hands and, subsequently, be misused. If a CAD model is not only a data storage medium, but becomes also a knowledge storage medium, the loss of it could implicate the loss of competitive advantage in

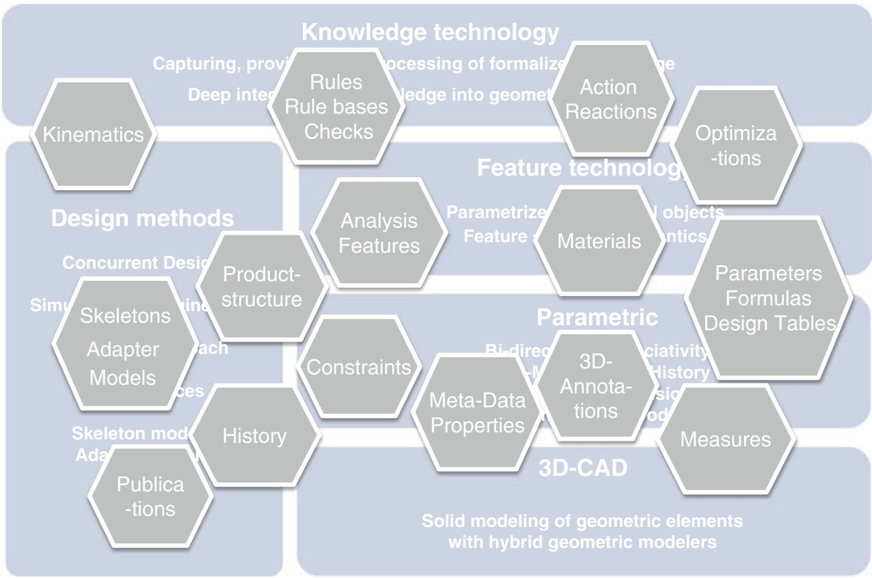


Fig. 18.4 Typical product knowledge items and design methods to be covered by IPP

the form of unintentional sharing of knowledge and, subsequently, the loss of a product innovation. Therefore, the flow of knowledge has to be controlled in a predefined and traceable way.

Looking again at Fig. 18.1, the next point of interest can be identified: the (built) product with all its components, including the non-physical components and spare parts. The oldest and apparently methodically easiest way for misuse of IPR is to rebuild a machine or a part of it by owning it and then using a more or less sophisticated method of reverse engineering (see Chap. 12) to acquire the know-how needed for production and assembly. In such a way plagiarism or counterfeit arises. To complicate or disable that route also belongs to the action fields of IP protection.

Finally, further explanation in this chapter will be classified—in accordance with the aforementioned technical scope and Fig. 18.1—in supporting methods for the protection of patents (Sect. 18.5), the protection of product data (Sect. 18.6), and protection of products and parts (Sect. 18.7). Considering its crucial importance and consequences on CE, the protection of product data which are generated during the product creation process will get the major weight.

18.5 Protection of Patents

Protection on a legal level includes measures to foster the patent maintenance process, avoid patent infringement and support patent litigation. For this purpose an IPDSS based on POE, technology clustering, and patent infringement analysis is developed [28]. First, the research develops a procedure for POE by analyzing the claims of the patent, the claim components, and the technical key phrases. Afterward, the research evaluates different key phrase extraction methodologies for searching related patents from relevant databases for USA (USPTO), Europe (EPO) and world (WIPO). The related patents include prior-art patents (applied before the application of the target patent) and the after-art patents (applied after the application of the target patent). The prior art patents are used to analyze the patent validity by using a modified overlapping clustering methodology for extracting the related technical characteristic of the patent documents. Finally, the research designs a patent infringement defense system based on the infringement identification rule (i.e., Literal Infringement, Doctrine of Equivalents). A domain ontology (see Sect. 10.2) is used to support the defense system objectively and systematically. Thus, the IPDSS process uses the proposed ontology, the key phrase evaluation and extraction methodology, advanced overlapping clustering, and the patent infringement rules as the bases of the methodology.

18.5.1 Patent Ontology Engineering

Based on a patent specification and a fundamental approach for ontology engineering (see Sect. 10.2), the proposed POE collects two types of patent information. One is the patent metadata including patent number, application date, issued date, inventor, assignee, international patent classification (IPC), U.S. patent classification, and reference information. The other type of patent information collected is the technical context including patent title, abstract, claims, background of invention, and detail description. A patent claim defines the limits of the rights granted by the independent claim and the dependent claim where the independent claims stand on their own, and the dependent claims depend on a single claim or on several claims and generally express particular embodiments as fall-back positions. The claims are written in a legalistic form, in which every claim starts with a capital letter (the only capital letter permitted in a claim) and ends with a period (the only one permitted in a claim), and each element in the claim must be named before it is used. Thus, the legalistic form of the claim is analyzed to identify the preamble, transition phrase, and body of the claim. The preamble sets forth the name and environment of the invention. The transitional phrases in patent applications are important, as the transition specifies whether the claim is limited to only the elements listed, or whether the claim covers items or processes that have additional elements. The body of a claim defines the elements or steps of the claim.

Patent infringement analysis involves basically two steps: claim construction, and comparison of the accused product to the properly construed claim. In the first step, which is exclusively a matter of law for the court, each asserted claim is construed to determine its scope and meaning. In the second step, a fact finder compares each properly construed claim to the accused device, to determine whether all of the claim limitations are present in that device, either literally or by a substantial equivalent.

This approach uses WordNet (a large lexical database of English) or Visual Thesaurus to search domain concepts or synonyms of technical terms. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct concept. The Visual Thesaurus is an interactive dictionary and thesaurus which creates word maps that blossom with meanings and branch to related words. After domain concept recognition, the technical key phrases are analyzed by the key phrase extraction methodology for evaluating the similarity of the related patents described below. R&D engineers finalize the claim construction using the metadata, the elements of the claims, the domain concept and the synonyms, and the technical key phrases describing the invention.

18.5.2 Patent Key Phrase Extraction

The patent analysis uses the IPC to collect the document set for limiting the scope of the analysis. A patent document contains different structures used to represent different meanings. Key phrases used in such document are from particular interest. The term-frequency (TF) approach is used to collect key phrases with high frequency in a patent document. Following to that, the inverse-document-frequency (IDF) approach is used to evaluate the frequency of key phrases that appeared in different documents. If a phrase appears in many documents, the value of the IDF would be low meaning that the phrase is not a good distinguishing representation of a document. Specific key phrases in patent analysis are eliminated using the IDF approach [28]. Further, the normalized approach of key phrases evaluation is calculated based on the lengths of the structured patent sections, such as title, abstract, claims, invention, and detailed description. This approach uses the normalized patent term frequency (NPTF) to analyze the value of the key phrases. Finally, instead of using the IDF approach as the distinguishing measure of key phrases, information gain (IG) is also employed as a term-goodness criterion. The IG approach measures the amount of information within a category by knowing the presence or absence of a phrase in a document.

18.5.3 Modified Overlapping Clustering

After completing the process of claim construction, POE, and the NPTF-IG key phrases extraction methodology, the target patent and the related prior art patents are analyzed to identify patent novelty. An invention is patentable if it satisfies specifications for novelty. Novelty is defined in Section 102 of the US Patent Law: *The invention is novel if it was not previously patented, described in a publication, in public use, or on sale by others before the inventor invented it*. Trappey et al. [35] use non-exhaustive overlap clustering to analyze patent documents with multiple technical descriptions of the invention. The results show that key phrases influence the similarity of the non-exhaustive clusters where an object is permitted to belong to no cluster. The initial clustering center is evaluated based on the value calculated by the crowding value (CV, maximum of data in a cluster) and the maximum distance value (MDV, maximum distance of each cluster). Non-exhaustive clustering may cause outliers to cause excessive overlapping results. Thus, this research uses hierarchical clustering such as the merging approach or the separating approach to improve the non-exhaustive clustering and is referred to as the modified overlapping clustering.

The research sets the clustering lower limit (CLL) to minimize outliers. If the data in a cluster is smaller than the CLL, the cluster is merged with the neighbor. The merging limit (ML) is used to avoid excessive overlapping. If the overlapping data set is larger than the ML between the two clusters, then the clusters are merged. Finally, the clustering upper limit (CUL) is set to control the size of the clustering result. If the amount of data in a cluster is larger than the CUL, the cluster is separated using k-medoids clustering [36]. The three parameters of the modified overlapping clustering facilitates to find prior art from related patents collected by the IPC scope and use the application date for checking the patent validity related to the target patent.

18.5.4 Case Study

The case study focuses on the Taiwan's Industrial Technology Research Institute (ITRI) litigation with LG Electronics (LGE) to claim patent infringement. In 2010, ITRI accused LGE, the world's third-largest mobile-phone maker, of infringing on 22 US patents on mobile phones, air conditioners, Blu-ray disc players and liquid crystal display (LCD) televisions. The complaints were formally submitted on Nov. 26 in U. S. Federal Court in Tyler, Texas [28]. ITRI claimed LGE was infringing on patents, including 15 related to LCD televisions and two related to mobile phones. ITRI filed four lawsuits against LGE (litigant case no. 6:2010cv00628, 6:2010cv00629, 6:2010cv00630, and 6:2010cv00631). This study chooses the US patent no. US5714247 litigated against litigant case no. 6:2010cv00629 as target patents for case analysis.

18.5.4.1 Patent Text and Data Mining

Using the developed (IPDSS—http://wheeljet.com.tw), the target patent number is set as TP(xi) and xi = US5714247 to download the document from USPTO. The extracted patent metadata is stored in the IPDSS's data-base as MD(xi). The developed system automatically analyzes 19 claims of the target patent which contain 3 independent claims as IC(xi) and 16 dependent claims as DC(xi) to help R&D engineer to interpret the scope of the intellectual property. Afterward, the system analyzes the structure of the independent claims by identifying the preamble, the transition phrases, and the body. The claim components of the body are analyzed by the antecedent basis as CC(xi). The single claim of the target patent is depicted in Fig. 18.5.

The key phrase analysis results (NPTF-IDF with NPTF-IG) are shown in Table 18.1. The IDF weight is calculated by the appearance of the document set, and the IG weight is calculated by the frequency of the specific domain. The result of the NPTF-IDF approach represents that the key phrase (e.g., light, Rate = 20 %) located in a few documents are the important phrases. The domain phrases appeared in all documents will likely be eliminated using the IDF's formula. Thus, the proposed NPTF-IG approach extracts the key phrases KP(xi) important to the

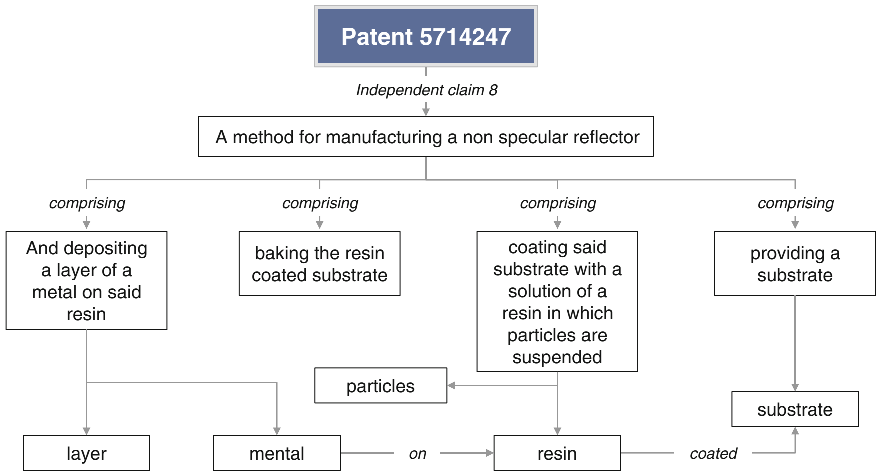


Fig. 18.5 The single claim of the target patent US5714247

Table 18.1 The appearance rate of the key phrases detected by NPTF-IDF and NPTF-IG

TOP	NPTF-IDF	Rate (%)	NPTF-IG	Rate (%)
1	KP98: light	20	KP1: substrate	100
2	KP60: bumps	40	KP4: liquid	100
3	KP37: electrodes	40	KP5: crystal	100
4	KP32: reflection	60	KP2: liquid crystal	100
5	KP29: insulating	60	KP24: formed	100
6	KP53: made	40	KP3: surface	100
7	KP20: reflector	60	KP7: layer	100
8	KP66: step	40	KP12: display	100
9	KP13: portions	60	KP9: reflective	80
10	KP62: thin	40	KP21: film	80
The appearance rate		46.4		87.2

specific domain. The key phrases extracted by the NPTF-IG approach is useful to evaluate the technical characteristics of the patent. Thus, the research proposes the NPTF-IG approach as the key phrase extraction methodology to analyze the POE.

The data of POE contains metadata MD(xi), independent claims IC(xi), de-pendent claims DC(xi), claim components CC(xi), component synonyms CS (xi), and key phrases KP(xi). The Microsoft Office Visio tool is used to build the patent ontology and is translated into standard XML in the IPDSS to analyze the patent context. The IPDSS then identifies potential patent infringement and evaluates patent validity.

18.5.4.2 Patent Litigation Strategy

After building the patent ontology, the target patent (US5714247) and its related patents are analyzed. The metadata of the patents, claims and key phrases are used to search the possible prior arts for patent validity. The search strategy for prior art is defined as PA(AD(xi), IPC(xi), UPC(xi), AN(xi), CC(xi), CS(xi), KP(xi)). For example, the result of the search rule (((ICL/G02F1$ AND CCL/349$) AND APD/19880101 ->19960614) AND ((ACLM/resin AND ACLM/substrate) AND (((ACLM/reflector OR ACLM/reflective) OR ACLM/reflecting) OR ACLM/reflect))) yields 23 patents as PA = $\{x1, x2, x3, \ldots, x23\}$. Based on the NPTF-IG extraction methodology and the modified overlapping clustering, the searched prior art and the target patents are used for building the key phrases matrix. The IPDSS can cluster the patents located in the group with the target patent as $\{y1, y2, \ldots, yk\}$ and $y1, y2, \ldots, yk \in \{x1, x2, x3, \ldots, xn\}$.

In the case study, the invention of the target patent represents a non-specular reflecting surface for use in a LCD and is obtained by randomly embedding particles in a layer of a resin solution and then baking to hardness. The particles' sizes are approximately the same as the layer's thickness, so a randomly uneven surface is the result. Based on the patent reference analysis, the prior art of the target patent are issued by Sharp Kabushiki Kaisha (US5418635 and US5408345). After the modified overlapping cluster analysis, the compared patent US5418635 is clustered with the target patent US5714247. Thus, the compared patent and the target patent are analyzed to find the same or similar components in the claims. First, the IPDSS finds the same reflective substrate from Claim 1 of the target patent and Claim 5 of the compared patent. Second, a method for manufacturing a reflective substrate is both extracted from Claim 8 of the target patent and Claim 1 of the compared patent. The claims of the compared and target patents are analyzed to find evidence of prior art.

If LGE holds patent US5418635 as a prior art, the target patent invalidity strategy should defend ITRI's litigation against LGE. When patent infringement litigation occurred, the litigated patents are analyzed for potential prior art identification. In this research, the IPDSS can help engineers analyze patents systematically using POE and prior art analysis. The modified overlapping clustering is also applied to support the R&D engineers effectively comparing the target patent and the most related patents.

18.5.5 Patent Quality Analysis and Applications

Patent quality analysis provides a means by which companies determine whether or not to customize and manufacture innovative products. Thorough patent research is needed to estimate the quality of patent documents. Novel approach is developed to improve the analysis and ranking of patent quality [37].

The first step is to collect technology specific patents and to identify relevant patent quality performance indicators. The second step is to identify the key impact factors using principal component analysis. These factors are then used as the input parameters for a back-propagation neural network model. Patent transactions help judge patent quality and patents which are licensed or sold with IPRs are considered high quality patents. The research method collects a set of patents sold or licensed and another set of patents unsold but belong to the same technology specific domains of interest. After training the patent quality model using these two sets of patents, other historical patents can be used to verify the performance of the trained model. The match between the analytical results and the actual trading status reached a good level of accuracy. In principle, the patent quality methodology evaluates the quality of patents automatically and effectively as a preliminary screening solution, which saves domain experts or researchers valuable time targeting high value patents for R&D commercialization.

Similar approach is used to analyze the Long-Term Evolution (LTE) technology development trend in the fourth generation (4G) mobile communication [38]. The LTE patent documents in USPTO database are searched. The research identifies 62 Wi-LAN patents as major target patents. These patents are analyzed with various patent statistic views (also called patent maps), such as patent counts based on IPC and forward citation counts. The high quality patents are categorized in technology-function matrix. The purpose is to identify the high quality patents in specific sub-technical or functional domains, which may lead to potential litigations in the future.

18.6 Protection of Product Data

Because it is easy to steal information during the exchange of product data, there is a need to develop processes, methods, and technologies to support the smooth introduction of IP protection and to ensure interoperability in cross-enterprise collaboration. Different basic methodological attempts to control the flow of information are focused on technical systems used to minimize the ways in which intellectual property used in information flow can be lost.

18.6.1 Setting-up the Protection Environment for a Collaboration Project

Having decided to collaborate, partners have to set up the corresponding collaboration processes and infrastructure as a mandatory component of their enterprise architecture. A connector must be provided as a "collaboration hub" which is based on modern IT technology using a highly customizable PDM system as its technological fundament. It fulfills two aims: seamless access and exchange of data which are created and are needed in project, and translation of corresponding data to the semantics of the end users. Processes and infrastructure must represent the temporary character of a project. For such a purpose, access through a project partner is necessary to the organization, network, system architecture (DMZ, firewall), data in system, and corresponding content.

The generic procedure of setting-up a collaboration hub is shown in Fig. 18.6. It is subdivided into 8 consecutive steps and starts with import of basic data and metadata, followed with definition of project template for access and selection of data. In the following steps collector mechanisms are activated, the different user access rights are set, and simulation of data import is conducted. Furthermore, it encompasses the major changes which may occur in project (update of project, change of project), and, finally, the project end or cancellation.

Once the collaboration hub is set-up, it can be used for daily data exchange as shown in Fig. 18.6. End use has two basic functions: push (provide the data for partner) and pull (get the data from partner). For each typical workflow (e.g. first draft, approval, change request etc.) a corresponding template is provided by collaboration hub to automate the working procedure. Such template encompasses all restrictions (e.g. digital rights management). Its fundamental component is a mapping engine [39] which translates the data content. The funnel at the bottom demonstrates how the data access and content can be flexibly adjusted to the project needs (Fig. 18.7).

A collaborative hub is an expensive solution which requires high initial investment and administration overhead. For such reasons it remains exclusive to

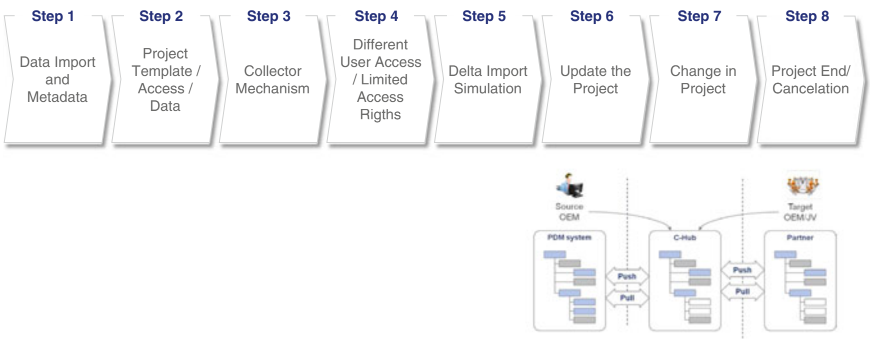


Fig. 18.6 Setting-up data protection in a collaboration by using collaboration hub (reproduced with kind permission from M. Fink © 2014)

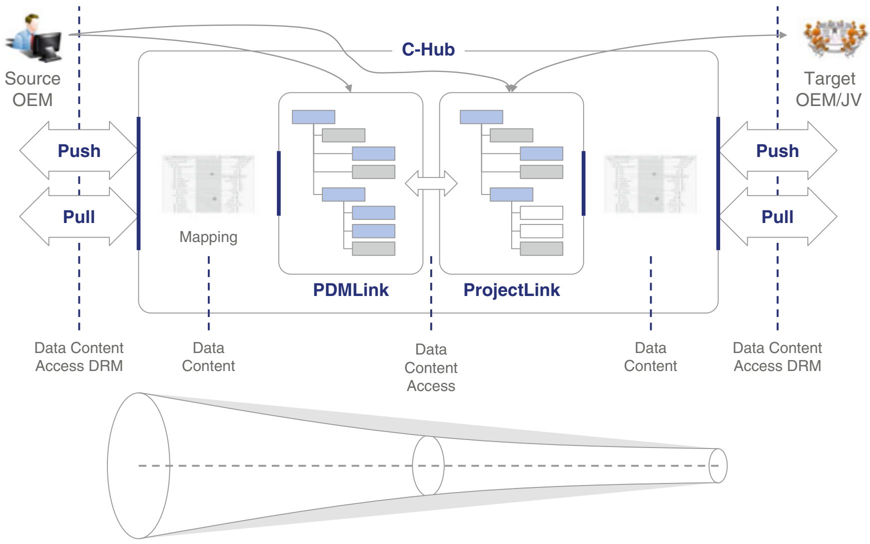


Fig. 18.7 Context diagram and data flow in collaboration hub (reproduced with kind permission from M. Fink © 2014)

large companies and large projects which can justify such a overhead (e.g. joint-ventures between automotive OEM). Otherwise, powered by its high interoperability, it can collect all singular approaches describes in the following sections.

18.6.2 Basic Approaches for Protection of Product Data

Providing solutions to achieve a high level of security in access and exchange of product data, in line with current practice, as well as requirements harmonization and methods development is the aim of ProSTEP iViP Association's Secure Product Creation Processes (SP2) Project Group, founded in 2007. In 2011 a successive project group was founded, the "Enterprise Rights Management (ERM) Open group" aiming to increase interoperability and simplify the integration of ERM solutions [40].

There are different technical approaches to combat the infringement of intellectual property. The following five practically relevant approaches have been identified initially in the SP2 project [41]:

- Data filtering to reduce the amount of know-how before distributing the data (Fig. 18.8a).
- Terminal server solutions that only display data to the recipient (Fig. 18.8b).
- Using Data Leakage Prevention to stop unauthorized data leakage over known interfaces (Fig. 18.8c)

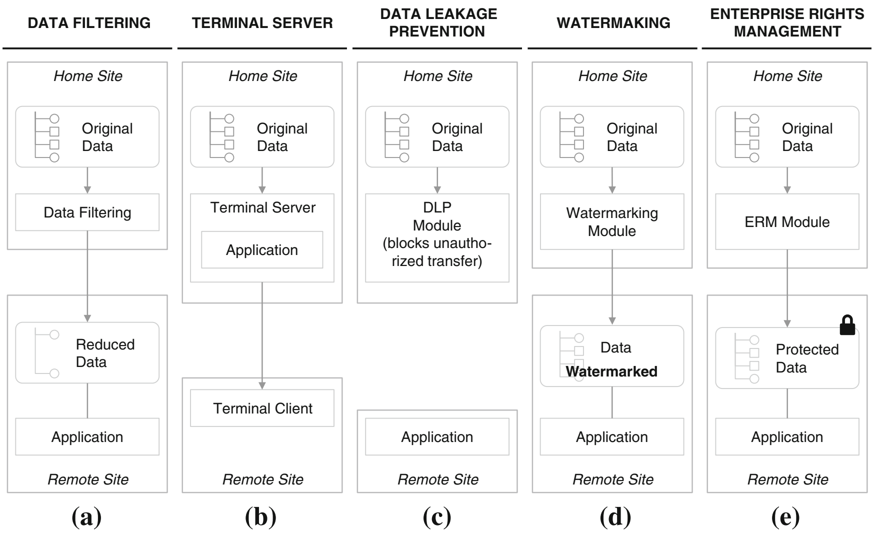


Fig. 18.8 Technical approaches to protecting intellectual property in product data

- Using watermarks to allow the identification of the source of a data leak (Fig. 18.8d).
- Using ERM to retain full control over distributed data, even after data exchange (Fig. 18.8e).

For OEMs and suppliers data filtering (model simplification) and ERM are the most important action fields (Fig. 18.8). ERM is more in the focus of an OEM which requires a hierarchical solution for the whole supply chain. The supplier is more interested in IP protection by physical model simplification because after data filtering it is not possible to track elements containing knowledge that were inside the product data before data transfer. The challenge for developing methods for IP protection is to keep up with the advancement of design methods and knowledge representations in CAD systems. Also organizational issues have to be taken into account.

The lack of appropriate standards for the interaction of singular solutions for IP protection has already been identified as a huge obstacle for the practical use of integrated IP protection solutions.

18.6.3 Solutions for Protection of Product Data

IP protection of product data can take place on the physical, content, and access right level [1].

The physical level supports protection against unauthorized access, addressing of the wrong recipient or physical loss of all (not only CAD) user data which leave the proprietary enterprise network for data exchange. Customary IT methods like authorization, authentication and encoding can be used.

Protection on the content level comprises intelligent filtering or intentional manipulation of CAD models prior to data exchange. It generates additional customer CAD models Derived from the original CAD models with reduced or minimal knowledge content compliant to relevant CAD data quality requirements.

Protection on the access right level controls access rights to product data in the context of a supply chain. It allocates access rights for certain users, for a certain period, and also for certain content, or inserts digital watermarks to demonstrate ownership and indicate tampering.

18.6.3.1 Protection on the Physical Level

For the implementation of data exchange in a collaborative environment secure platforms are commercially offered today on the market. Those platforms implement methods and solutions for IP protection based on encryption and portal technologies.

A comprehensive coverage of technical solutions is covered (Fig. 18.9) that fulfill different needs of a holistic IPP solution. Such data exchange platforms enable the secure, stable and easy exchange of all data via the Internet in a virtual enterprise. They offer a highly secure encryption concept and are optimized for the stable transfer of large user data volumes. Some of their key functions are: High level of data security via encryption, integrated public key infrastructure (PKI), stable and quick transmission of large data volumes, flexible and open structures that allow a large number of processes to be automated and scalable solution modules tailored to individual needs. As an example, the OpenDXM GlobalX is described in the use case of a manufacturer of sport cars [42].

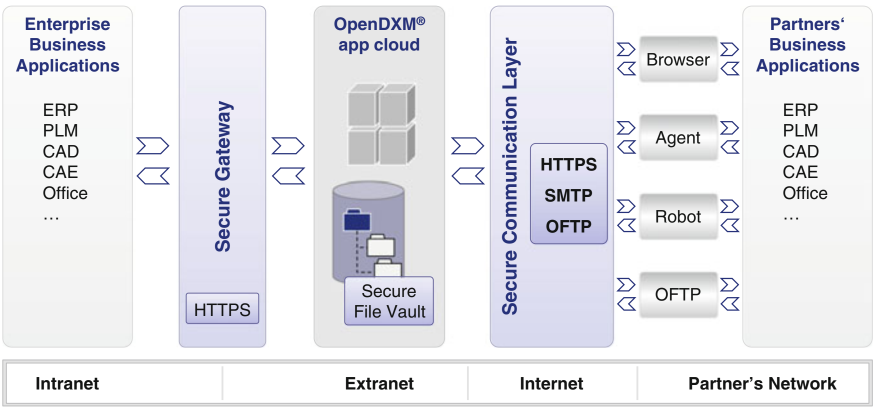


Fig. 18.9 Context diagram of secure data exchange platform [42]

18.6.3.2 Protection on the Content Level

In [39], a comprehensive methodical attempt is described to take the determining factors of the data exchange relationships described above in order to selectively filter the data to the minimal content that satisfies the data recipient (Fig. 18.10). In other words, IP protection is always driven by the requirements of the respective process chain (design in context, DMU etc.). For example, many companies (e.g., 1st-Tier Suppliers) are applying different data exchange agreements with their customers (e.g. OEMs). The data exchange agreements differ with respect to the content and purpose of CAD data to be delivered, i.e., the degree of "parameterization" and the data exchange frequency. It is very important to be able to remove dedicated knowledge portions from the CAD data very precisely and context-sensitively to satisfy the different data exchange agreements (Fig. 18.10). The main goal is very often to deliver minimal knowledge with the CAD data, but to satisfy the data exchange and data quality thresholds. Depending on the use case there is a strong need to define some IPP rules for controlling the IPP process in a predefined way.

Based on the background of KBE projects and customers' needs for IPP software solutions for IPP have been developed in the last years. These solutions are currently in use in the companies of many industrial customers. These project-driven implementations of IPP software are known by the term "Knowledge Editor" [43] and work similarly to an additional CAD workbench.

IPP methods have been implemented for CATIA V5 and Creo dedicated to manage design knowledge in CAD models (Fig. 18.11). The respective software "Knowledge Editor" is used for analyzing data with respect to valuable product knowledge and is used to prepare the data before transfer, i.e., removal or change of dedicated items. The comprehensive but also selective removal of items is controlled by predefined IPP templates.

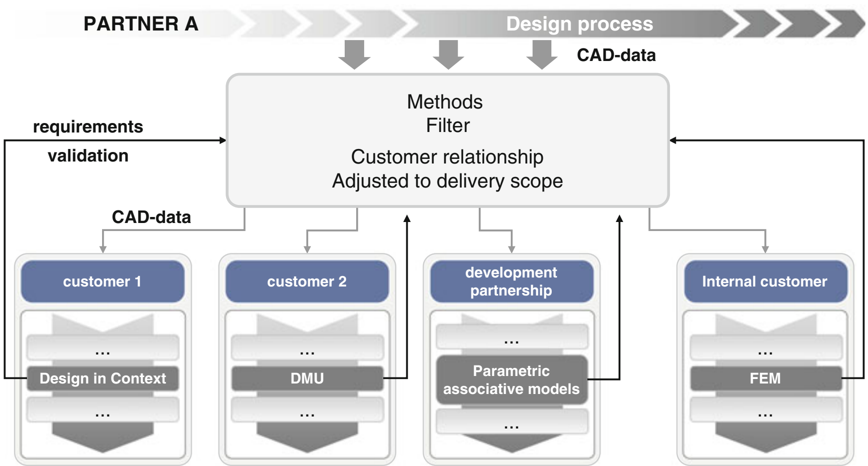


Fig. 18.10 Process chain-driven data preparation

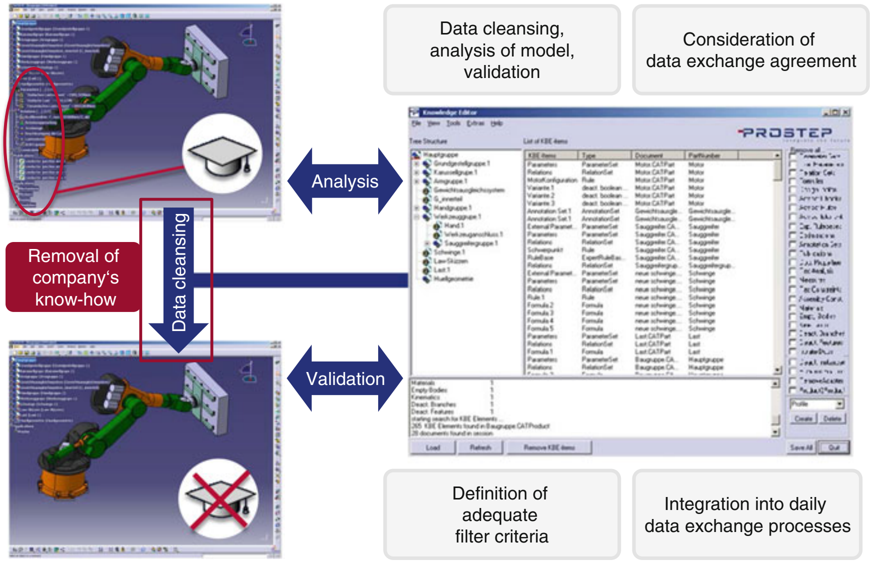


Fig. 18.11 Knowledge editor

IPP templates for flexible tailoring of CAD data filtering can be easily defined by the end user with the "Rule Assistant", a wizard-like extension of the Knowledge Editor. The end user defines rule bases (Fig. 18.12), which can be automatically processed by the Knowledge Editor. With the defined rules, it is possible to process a very comprehensive knowledge removal in order to retain a minimal IP content in the CAD data but still to meet the customer's data exchange requirements. For example, it is possible to delete some knowledge but to preserve special KBE elements, e.g., in templates and start-up models, or to combine model simplification with product structure shading. Also the user can prepare CAD modes for special collaboration scenarios. As an added value the Rule Assistant can be used for merging CAD content into OEM start models which covers the basic setup of CAD models created in an CAD environment.

After knowledge and feature removal the Knowledge Editor applies some validation methods to assure the defined CAD data quality. In all cases the process of knowledge removal is secure and non-reversible. Later on, the data recipient will not be able to trace knowledge items previously included in the model.

The Knowledge Editor-based multi-level processes have to be managed and automated to be suitable for the daily data exchange of a company. The integration of software-based IPP in day-to-day data exchange processes can be easily done, e.g., through the use of OpenDXM® GlobalX as described in [38]. Thus, IPP is reduced to a manageable, repeatable process step in the data exchange chain with transparent monitoring and result evaluation. Data preparation can even be done for multiple recipients with multiple process requirements and filter criteria. After introduction the customer can operate the IPP solution for his purpose [43].

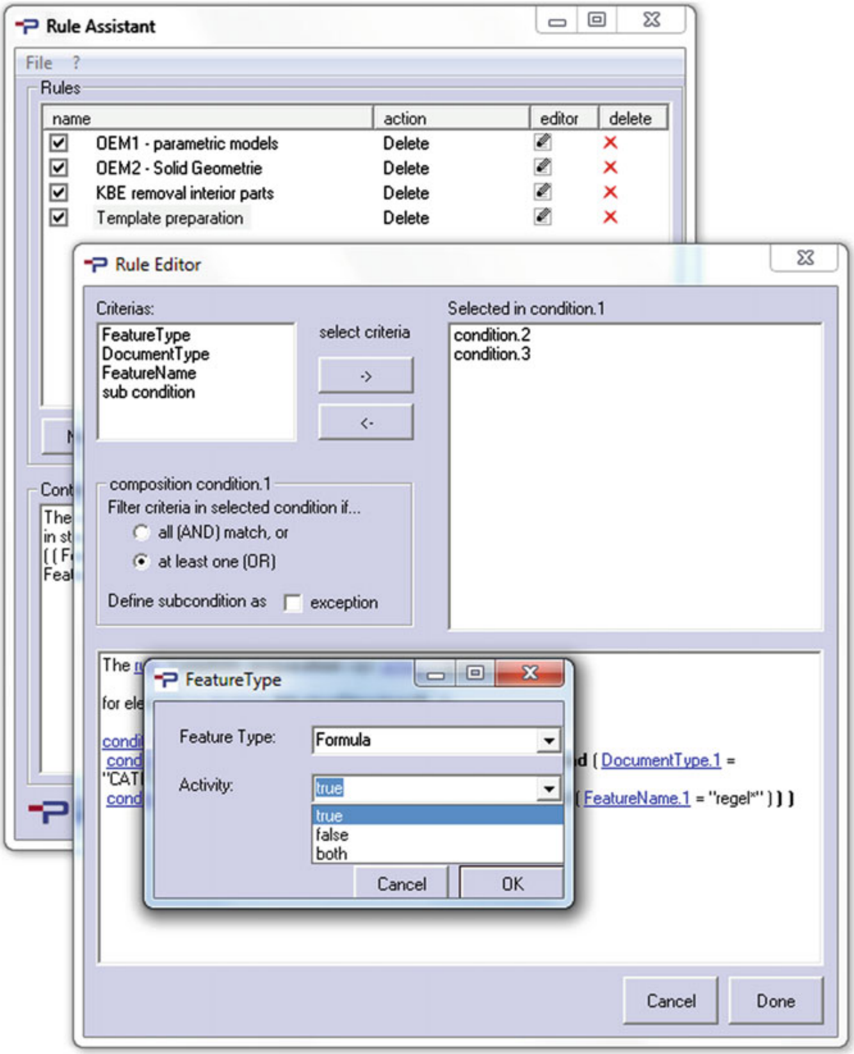


Fig. 18.12 Rule assistant

Another solution approach subdivides the CAD model that needs to be exchanged in two portions: a confidential portion which has to be encoded, and a non-confidential portion which can be fully accessed by a partner without any restriction. The subdivision of the free form surface description in such a way can be realized with wavelets [44]. These functions are used to divide the data into hierarchically arranged coefficients. A crucial step is to localize confidential information in the coefficients of one or a few levels and encode these contributions only. The selection of the usable wavelets is an important choice, because it influences the possibilities and limitations of the whole method. As a result a semi-destroyed model is created that contains slightly changed geometry in the sensible area besides the original geometry in other area. This solution is used at Hella KG Hueck & Co., the large manufacturer of automotive headlights, to protect IP stored in the geometry of reflectors in daily business. Such solutions can be easily adapted to the automatic procedure for model preparation as depicted in Fig. 18.10.

18.6.3.3 Protection on Access Rights Level

For IP protection on the access rights level methods like digital rights management (DRM) and digital watermarking are developed [45]. DRM has been already state-of-the-art for many applications for years, e.g., Adobe Digital Enterprise Platform (ADEP), which is widely known by its use in "intelligent PDF documents" in business communication. It is expected that this software suite will be extended to handle CAD files, too.

For this purpose the PDF Generator 3D has been developed as a server technology fully integrated into the ADEP (Fig. 18.13). It combines 3D-geometry, PDM/ERP meta data and interactivity in a lightweight PDF document shareable throughout design and beyond firewalls. To use the CAD models translation into specific U3D/PRC format is necessary which is embedded into PDF document. The embedding of other 3D formats like JT is also possible and has been proven in practice. With this solution IPP is performed by controlled access rights to PDF documents with embedded 3D models [46].

Despite many attempts and research projects nobody yet succeeded to apply the DRM/ERM to singular entities within a CAD model until now. Such IPP methods and tools are still subject to further research while other (see Sects. 18.6.1 and 18.6.2)

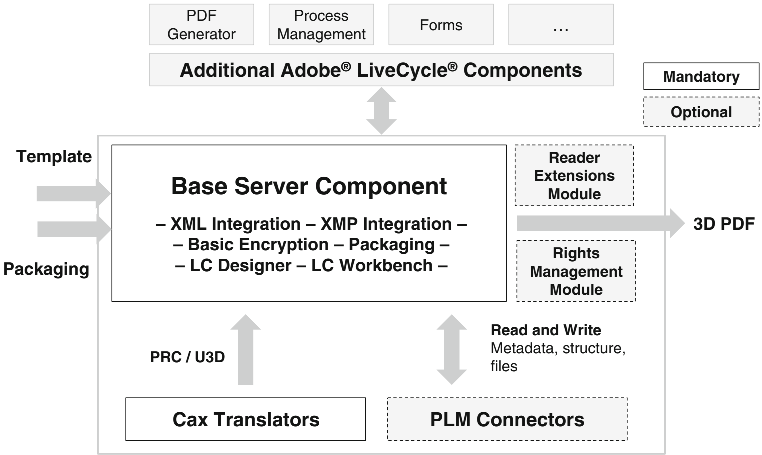


Fig. 18.13 Context diagram PDF generator 3D

have already been implemented and well-adjusted to the requirements of the respective information flow. The aforementioned project group ERM.Open is still working on standardization of ERM.

As an additional method digital watermarking is frequently used for visible or invisible marking of images that serve ownership demonstration and tampering verification [47]. This method can be used for marking of NURBS representation in CAD models too where the reversible, invisible digital watermark is composed of a combination of a robust watermark with one bit information content and a semi-fragile watermarks with variable information content [48]. The information obtained from the semi-fragile watermark can be used to remove the robust watermark from the model and then restore the original geometry. The predefined tolerance threshold (e.g., manufacturing tolerances) in the NURBS representation of the original models can be accurately maintained. The properties of the NURBS representation are exploited for preserving the continuity between adjacent patches because the continuity is the major criterion of designers for assessing the quality of surface models. Robustness limits of the watermark have been successfully evaluated for two typical processing operations of NURBS-based CAD models, tessellation and translation between different modeling kernels.

18.7 Protection of Products and Parts

Considering measures for IP protection of products and parts it is necessary to distinguish between tangible and intangible (e.g. software) products as well as first shipment and spare parts. In general, authentication of products is necessary to avoid the intentional or non-intentional use of counterfeits.

18.7.1 Tracking and Tracing

The main objective of tracking and tracing is to complicate and avoid counterfeiting of critical components and spare parts of complex machines and equipment through integrated protection by marking and authenticating products at dedicated points in the value added chain during product exploitation [49].

The procedure tracking and tracing is the following:

- Identify critical parts and components that need to be marked by an OEM.
- Select a suitable marking technology for parts and components.
- Mark selected parts and components.
- Design, implement and run a distributed IT system to track and trace the marked products within the value-added chain directly by the OEM or by an authorized service provider.

In an enterprise threatened by counterfeiting and piracy, adding security markings to every manufactured part and component, to always be able to recognize them as original, would be cumbersome and expensive, because marking and later checking each single product cause additional expenses. Therefore, it is necessary to directly identify the parts which are most in danger of being counterfeited and are also of special interest to the original manufacturer.

After the parts and components to be protected have been identified, it is necessary to select a suitable marking technology for each of them. This can include technologies like holograms, infrared colors (IR), copy detection patterns (CDP) and radio frequency identification (RFID). The first two technologies are denoted as originality marking technologies and the second two as unique marking technologies.

To guarantee the continuous tracking and tracing of marked parts a distributed IT system needs to be designed and implemented that includes all stakeholders (OEM, agent, customer) (Fig. 18.14). The identity of parts with worldwide unique markings can be read by a user at every identification point and stored together with the location and time stamp in a central data base. By using GPS the machines and equipment can always be exactly located, too.

Additional components and spare parts are particularly affected by counterfeit and piracy in the machinery and equipment sector. To support a customer running a machine, suitable readers can be installed as an identification point inside of the machine to verify that the machine only contains original components. This leads to significant advantages for the machine owner, because he can be sure of the quality and functionality of the original parts and doesn't risk losing the machine's warranty.

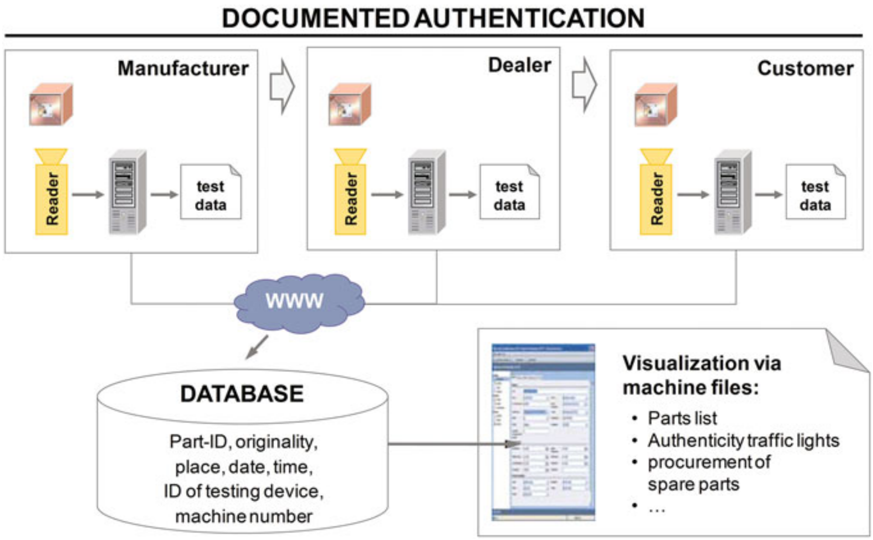


Fig. 18.14 Product tracking and tracing within the value added chain (reproduced with kind permission from D. Stockenberger © 2014 [49])

18.7.2 Protection of Embedded Software

Embedded systems are an important component of many modern products giving it added functionality and value. However, sophisticated techniques allow attacks on hardware and software in embedded systems, if they are not specifically protected. Numerous examples of complete reverse engineering and piracy are known. Since the protection of the software alone against the reverse engineering is hardly reliable, the protection measures shall include both the hardware and the software [50, p. 33].

With a software protection dongle as a hardware security module software can be protected against unauthorized duplication. In a separate memory a modern dongle often contains the necessary drivers along with cryptographic methods. The cryptographic keys are protected both against physical (e.g. side-channel attacks) and software-based attacks. Through use of encryption, it is difficult for an attacker to locate and disable the queries to the dongle hidden in the software.

Establishing a procedure for challenge-response authentication (Fig. 18.15) the dongle presence can be checked. Even a cryptographic dongle can be used during object time to protect the software against tampering (such as a cryptographic integrity check of the binary instructions in memory), while reverse engineering of the software becomes complicated. Therefore, a well implemented current dongle generation is still highly resistant.

A Secure Memory Device works like a cryptographic dongle. Instead of being connected to an external interface, a secure memory device is already integrated as a module with in hardware design. In addition to internal memory, which also contains individual parameters or may have a unique chip identifier, these devices provide useful basic cryptographic functions for protection of the entire system (hardware and software) (Fig. 18.15). If a secure memory device is built stationary in a component, the protection of the whole product is improved.

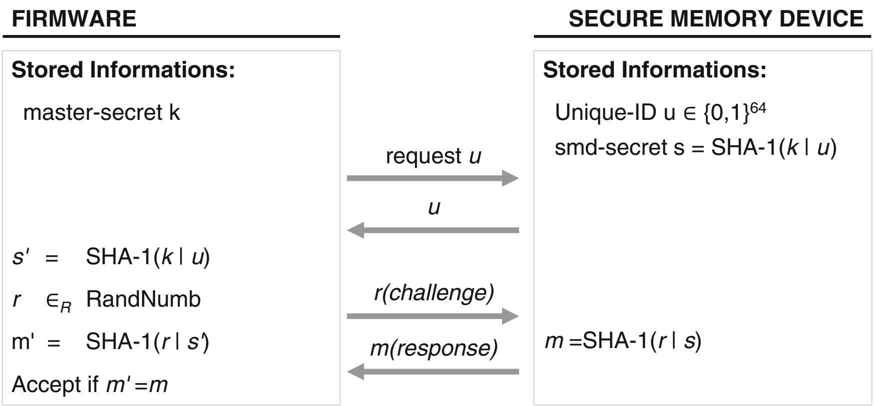


Fig. 18.15 Challenge-response authentication for presence check of hardware device (reproduced with kind permission from B. Filipovic © 2014 [50])

Protection of embedded software is a key components of a holistic protection concept based on systems engineering approach [51] considering the complete security life cycle of products and services starting with requirements and finishing with decommissioning.

18.8 Conclusions and Outlook

This chapter has presented the state of the art of implementing IPP in the context of CE. It underlines the emergence of IPP in a turbulent, globalized economy. It has focused on protection of rights, product data and final products and spare parts against counterfeiting in the various stages of a product life cycle [16].

An IPDSS and its methodology have been explained. The claim construction is used to process a patent ontology. Then, patent validity is analyzed by the proposed NPTF-IG key phrase extraction and the modified overlapping clustering method for prior arts finding. Finally, the IPDSS is implemented to support engineers evaluating the patent litigation systematically and effectively. IPDSS can automatically analyze the types of patent claims (independent or dependent claims) and the structure of the claims.

The protection of product data with high impact on CE has been described starting with a set-up of the protection environment for a collaboration project. Different levels are distinguished by definition of a methodical and repeatable process for CAD-model protection. Despite the recognition among expert practitioners of 3D-CAD systems as tools that efficiently improve individuals' and organizations' problem solving capabilities, their wide dissemination with full functionality is still limited by the risk of knowledge loss. Significant efforts are required to protect intellectual property (IP) and to prevent the misuse of product data. The described solution significantly reduces actual risk concerning the wide usage of KBE technology, thus facilitating the adoption of KBE for large scale industrial use.

Protection of products against plagiarism and counterfeits has been demonstrated on tangible and intangible products (embedded software).

The described solutions discover various benefits. First, they help to establish a secure collaboration and provides a way to protect intellectual property in a predefined and repeatable way. Secondly, they fulfill prerequisites for a wide use of KBE technology by increasing trust in internal processes and methods. Furthermore, they integrate IPP into the daily business. Thirdly, they help to save costs by automating the data exchange processes. Finally, they give protection against counterfeiting.

Further work would include the enhancement of singular components to work with higher granularity. Most efforts would have to be undertaken to improve DRM/ERM. Further standardization of IPP-related interfaces is also necessary. Finally, validation of the scalability of the approaches for large assembly sets as well as visualization of large relationship graphs remain as further objectives.

The potential impact of intellectual property assets is so large that it certainly will have a considerable effect on national and international economic development in the future [52]. A new interdisciplinary research field arises that combines economic and engineering insights, measurement approaches, and methodologies to ask fundamental questions concerning the viability of a free and open information society and to provide answers where possible [53]. Looking from the global perspective, three intersecting subjects with diverse emphases emerge [54]:

(1) whether strengthening protection of intellectual property stimulates or hinders technological learning and innovation in developing countries;
(2) ways in which knowledge is generated and transformed into useful technology for markets, that is to say, ways in which national innovation systems work;
(3) the role of public policy as an instrument for innovation and for regulation of intellectual property.

References

1. Liese H, Rulhoff S, Stjepandić J (2013) Enhancing product innovation by implementing intellectual property protection into the virtual product creation. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 267–278
2. Dyer JA (2014) The contest of the century: the new era of competition with China—and how America can win. Alfred A. Knopf, New York
3. Herman BD (2013) The fight over digital rights: the politics of copyright and technology. Cambridge University Press, New York
4. Atkinson B, Fitzgerald B (2014) A Short history of copyright. Springer International Publishing, Switzerland
5. Azevedo ML, Silva ST, Afonso O (2012) Intellectual property rights and endogenous economic growth—uncovering the main gaps in the research agenda. In: Teixeira AAC (ed) Technological change. InTech, Rijeka, pp 45–64
6. Stuiber P (2006) Doppelmayrs wundersame Seilbahnvermehrung in China, Die Welt, 17 Feb 2006. http://www.welt.de/print-welt/article198743/Doppelmayrs_wundersame_Seilbahnvermehrung_in_China.html. Accessed 22 March 2014
7. NN (2014) Apple Inc. v. Samsung Electronics Co., Ltd. Wikipedia. http://en.wikipedia.org/wiki/Apple_Inc._v._Samsung_Electronics_Co.,_Ltd. Accessed 15 March 2014
8. NN (2013) USA wollen stärker gegen Industriespionage aus China vorgehen, Portal financial.de, 21 Feb 2013. http://www.financial.de/news/agenturmeldungen/roundup-usa-wollen-starker-gegen-industriespionage-aus-china-vorgehen/. Accessed 23 March 2014
9. Neukirch R et al (2013) Das chinesische Problem. Der Spiegel 9, 25 Feb 2013. pp 20–22. http://www.spiegel.de/spiegel/print/d-91203384.html. Accessed 23 March 2014
10. NN (2013) Cyber Security Report 2013. Ergebnisse einer repräsentativen Befragung von Abgeordneten sowie Führungskräften in mittleren und großen Unternehmen. Deutsche Telekom/T-Systems. http://www.cybersecuritysummit.de/assets/downloads/131106_Cyber_Sicherheitsreport_2013_final.pdf. Accessed 15 March 2014
11. Richter R, Streb J (2009) Catching-up and falling behind knowledge spillover from American to German machine tool makers. Discussion Paper 09-2009, University of Hohenheim. https://fzid.uni-hohenheim.de/71978.html. Accessed 15 March 2014

12. Gragido W, Pirc J (2011) Cybercrime and espionage: an analysis of subversive multivector threats. Elsevier, Burlington
13. Philpott J, Jolly A (2004) A handbook of intellectual property management: protecting, developing and exploiting your IP assets. Kogan Page, London
14. Wong T, Dutfield G (2010) Intellectual property and human development: current trends and future scenarios. Cambridge University Press, Cambridge
15. Smallwood RF (2012) Safeguarding critical e-documents: implementing a program for securing confidential information assets. Wiley, Hoboken
16. VDMA (2013) Leitfaden zum Produkt- und Know-how-Schutz: Einführung von Schutzmaßnahmen gegen Produktpiraterie und Know-how-Abfluss. VDMA, Frankfurt
17. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manag 7(2):101–115
18. Bondar S, Ruppert C, Stjepandić J (2014) Ensuring data quality beyond change management in virtual enterprise. Int J Agile Syst Manag 7(3/4):304–323
19. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manag 6(4):307–323
20. Fuchs HJ et al (2006) Piraten, Fälscher und Kopierer. Gabler Verlag, Wiesbaden
21. Abele E, Kuske P, Lang H (2011) Schutz vor Produktpiraterie. Springer, Berlin
22. Quandt S (2006) The insider treat benchmark report—strategies for data protection. White paper, Aberdeen Group, Jan 2006. http://www.aberdeen.com
23. Jolly A (2010) The innovation handbook: how to profit from your ideas, intellectual property and market knowledge, 2nd edn. Kogan Page, London
24. McCarville MA (2012) Intellectual property, patents and innovation in aeronautics. In: Young TM, Hirst M (eds) Innovation in aeronautics. Woodhead Publishing, Cambridge, pp 263–303
25. Meiwald T (2011) Konzepte zum Schutz vor Produktpiraterie und unerwünschtem Know-how-Abfluss. PhD thesis, TU Müchen
26. Bader MA, Gassmann O, Ziegler N (2014) Managing the intellectual property portfolio. In: Gassmann O, Schweitzer F (eds) Management of the fuzzy front end of innovation. Springer International Publishing, Switzerland
27. Bryer LG, Lebson JS, Asbell MD (2011) Intellectual property operations and implementation in the 21st century corporation. Wiley, Hoboken
28. Wu CY, Trappey AJC, Trappey C (2012) Using patent ontology engineering for intellectual property defense support system. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering, Springer, London, pp 207–218, 2013
29. Hopgood AA (2011) Intelligent systems for engineers and scientists, 3rd edn. CRC Press, Boca Raton
30. Liese H (2004) Wissensbasierte 3D-CAD repräsentation. TU Darmstadt PhD thesis, Shaker Verlag, Aachen
31. Liese H, Stjepandić J (2004) Konstruktionsmethodik: Wissensbasierende 3D-CAD-Modellierung, CAD/CAM Report. Dressler Verlag, Heidelberg, 10. http://www.prostep.com/de/prostep/medien/cad-cam_kbe.htm. Accessed 15 March 2014
32. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manag 7(2):116–131
33. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3/4):324–347
34. Ong SK, Xu QL, Nee AYC (2008) Design reuse in product development modeling, analysis and optimization. World Scientific Publishing, Singapore
35. Trappey CV, Trappey AJC, Wu CY (2010) Clustering patents using non-exhaustive overlaps. J Syst Sci Syst Eng 19(2):162–181
36. Han J, Kamber M (2006) Data mining: concepts and techniques, 2nd edn. Morgan Kaufmann Publisher, San Fransisco

37. Trappey AJC, Trappey CV, Wu CY, Lin C-W (2012) A patent quality analysis for innovative technology and product development. Adv Eng Inform 26(1):26–34
38. Trappey AJC, Chen LWL, Chang JYC, Yeh MFM (2014) Strategic development of LTE mobile communication technology based on patent map analysis. In: Cha J et al (eds) Proceedings of 21th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 441–450
39. Yang J, Han S, Grau M, Mun D (2009) OpenPDM-based product data exchange among heterogeneous PDM systems in a distributed environment. Int J Adv Manufact Technol 40(9–10):1033–1043
40. ProSTEP iViP Association (2008) Project Group "Enterprise Rights Management (ERM. Open)". http://www.prostep.org/en/projects/enterprise-rights-management-ermopen.html. Accessed 15 May 2013
41. Spitznagel P, Liese H (2008) Intellectual property protection for CAD design. In: Proceedings of ProSTEP iViP Symposium 2008
42. Wendenburg M (2012) Data exchange on the fast track, CAD/CAM Report Engineering. Hoppenstedt, Darmstadt, 6. http://www.prostep.com/en/mn/press/prostep-in-the-media/detailed-news/artikel/299/data-exchang.html. Accessed 15 May 2013
43. Liese H, Rulhoff S, Stjepandić J (2010) Securing product know-how by embedding intellectual property protection into the organization. In: Proceedings of 16th international conference on concurrent enterprising, Lugano, 21–23 June 2010
44. Nawotki A (2001) Eine selektive Methode zur Verschlüsselung von Konstruktionsdaten mit Wavelets. PhD thesis, TU Kaiserslautern
45. Yu F, Lu Z, Luo H, Wang P (2010) Three-dimensional model analysis and processing. Springer, Berlin
46. Pfalzgraf P, Pfouga A, Trautmann T (2012) Cross enterprise change and release processes based on 3D PDF. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering, Springer, London, pp 753–763, 2013
47. Wang FH, Pan JS, Jain LC (2009) Innovations in digital watermarking techniques. Springer, Berlin
48. Funk W (2008) Digital watermarking for NURBS-based 3D CAD models. TU Darmstadt PhD thesis, Logos-Verlag, Berlin
49. Stockenberger D (2014) Schutz vor Produktpiraterie durch Kennzeichnung und Authentifizierung von Komponenten und Ersatzteilen im Maschinen- und Anlagenbau. PhD thesis, TU München
50. Filipović B, Schimmer O (2012) Protecting embedded systems against product piracy. Technological background and preventive measures. Fraunhofer AIESEC, Garching
51. Jacobs S (2011) Engineering information security: the application of systems engineering concepts to achieve information assurance. Wiley, Hoboken
52. Fussan CJ et al (2010) Managementmaßnahmen gegen Produktpiraterie und Industriespionage. Gabler Verlag, Wiesbaden
53. Böhme R (ed) (2013) The economics of information security and privacy. Springer, Berlin
54. Martínez-Piva JM (ed) (2009) Knowledge generation and protection: intellectual property, innovation and economic development. Springer, New York

Part III
Applications

Chapter 19
Challenges to Digital Product and Process Development Systems at BMW

Dietmar Trippner, Stefan Rude and Andreas Schreiber

Abstract Today, the methods of model based product development are well-recognized and wide spread, at least, in the automotive industry as well as in the aerospace industry and their suppliers. But, current challenges of these industries like light weight design, electro mobility, modern mobility concepts plus those caused by rising product complexity bring this concept to its limits. An overall approach is progressively requested, which is able to continuously integrate requirements, functions, logic and physical product descriptions (RFLP). This should be possible not only for mechanical aspects but also for electronics and software development. The approach of system engineering addresses the continuous availability and linkage of product information. This concept, which is well-known in the aerospace industry for a long time, is only recently used in automotive industry. An example is the use of integrated development environments. Nonetheless, the realization of this concept in an automotive company is definitely a challenge. Examples for these problems are differently coined. Examples are detailed requirements (client requirements versus requirements to a complete vehicle and to components properties), consideration of configuration, validity and maturity, complexity management (complete vehicle to component, vertical integration, plus integration of early concept phases over development, verification, clearance to the production start-up, horizontal integration) and multi-disciplinarity (mechanics with calculation, electronics and software). The realization of systems engineering does not only create high demands to the design of the process-IT (authoring systems, TDM and PDM), but also has to consider organizational aspects (process and structure organization, integration of development partners and suppliers). Frequent acquisitions under IT system vendors, especially, in the

D. Trippner (✉)
dreiconsult GbR, Feldstr. 8, 82290 Landsberied, Germany
e-mail: dr.trippner@dreiconsult.de

S. Rude · A. Schreiber
BMW AG, Munich, Germany
e-mail: stefan.rude@bmw.de

A. Schreiber
e-mail: andreas.as.schreiber@bmw.de

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_19

CAD/PLM/CAE market as well as the selection of the systems for functional and economical aspects lead to increased requirements concerning open interfaces. In the present document, findings and experiences from the introduction of systems engineering for automotive processes are described. Effects on the process IT architecture are outlined. “Lessons learned” and necessary changes in process-IT, in form of selected examples and solution alternatives, are discussed.

Keywords Systems engineering · PDM · TDM · CAE · Simulation · RFLP · Code of PLM openness (CPO)

19.1 Introduction

Digital product and process development systems have become indispensable for the permanent advancement and new development of products from the product program of BMW. Coming from the development of aircraft engines, motorcycles and racecars, the current product program of the brands BMW, Mini and Rolls Royce is presently enhanced massively in direction of electro mobility and mobility services (Fig. 19.1).

The premium strategy is consequently pursued in the different vehicle classes. By doing so, besides the classic car bodies also new variants are offered, for example like SACs, and all variants are offered again with different extra and country specific equipment (Fig. 19.2).

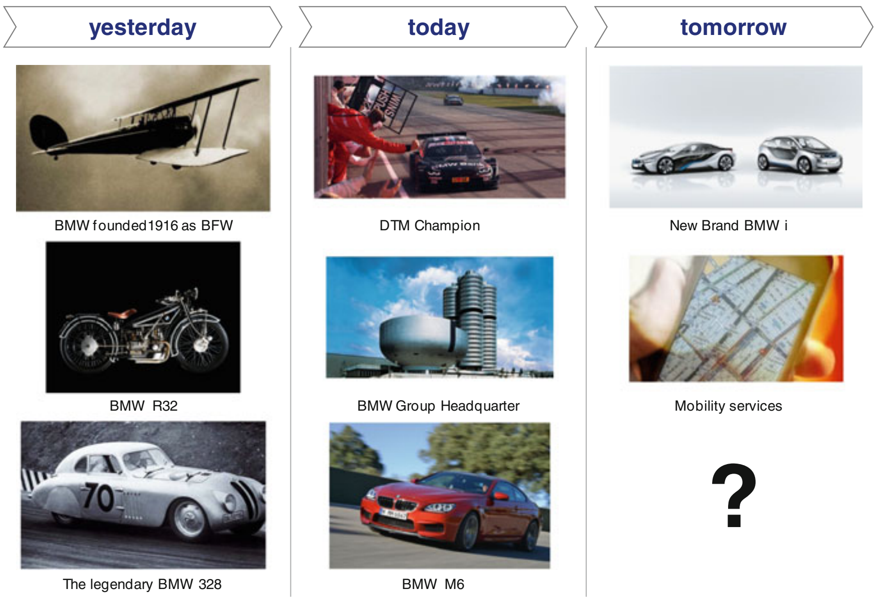


Fig. 19.1 BMW group—company portrait

Fig. 19.2 The BMW group product portfolio

Beside the growth of the product program, the possibilities of electronics lead to exponential gains of the vehicle functions, which are more and more realized as linked, software based functionality. Secondary, to those for the client facing functions, is the multiplicity also caused by increasing safety requirements of the vehicles and the improved client service. Additionally, the realization of multi-faceted requirements, for example to meet legal regulations, is relevant for the increasing complexity (Fig. 19.3).

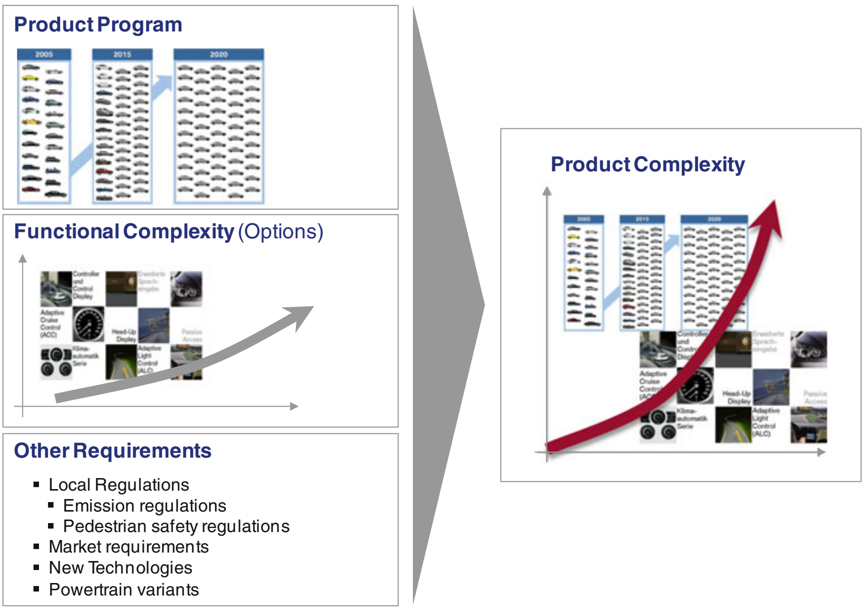


Fig. 19.3 Increased product complexity in the product development process

19.2 Requirements Caused by the Development Process

To control the multitude of requirements for the development of the complex product programs a highly optimized development methodology is necessary (Fig. 19.4).

Requirements for vehicles are described and modeled in the requirements management. The derivation of requirements from subsystems, which realize single functional groups, are going to be standard for different component or system suppliers. Still, BMW differs in geometrical design, functional design and system design. Functional design is reflecting the vehicle physics (properties like acoustics, oscillation or crash behavior), while system design is seen as the definition of the E/E systems inclusive their realization in vehicle software. Here, simulation methods already play a role, especially in the components design process.

Geometrical/functional integration as well as system integration serve for the gradual assembly of components, subsystems and complete vehicles. A gradual concept serves for the execution of necessary verification methods (test and validation) on a cost effective level. By virtual means as well as on the base of physical prototypes verification is understood as validation against specified requirements. The validation process checks against client expectations until the final approval of the complete vehicle.

As a base for the resulting design process the V-model, which is known from systems engineering, is suited (Fig. 19.5).

Development process requirements have to be defined as properties and functions. The gradual definition of subsystems and components is called architecture

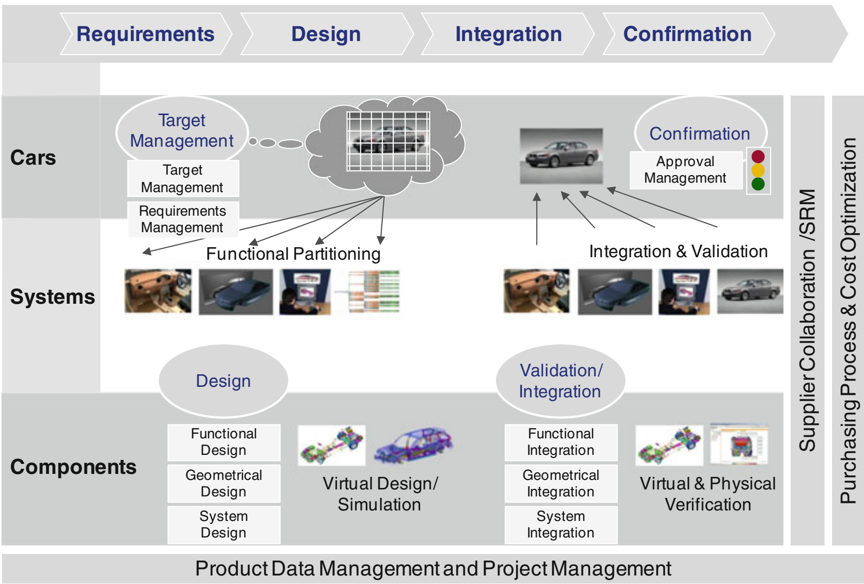


Fig. 19.4 The principles of the product development process

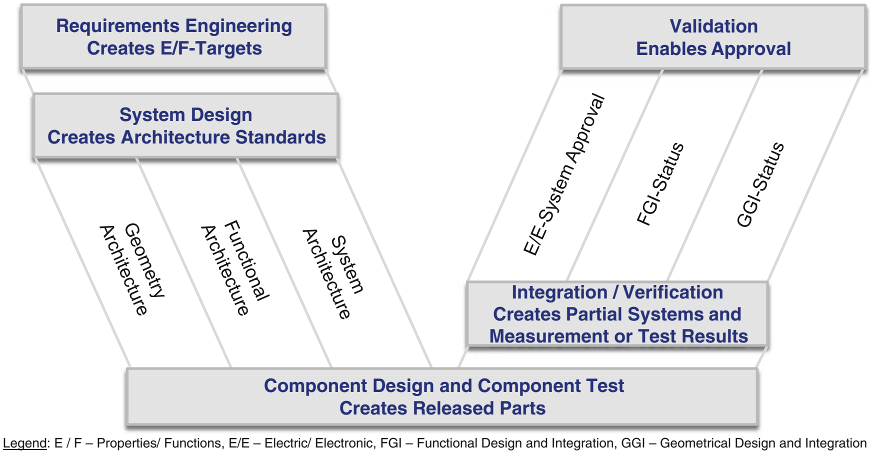


Fig. 19.5 Development process in the V model

development, which has a different interpretation in different disciplines. Specifications for the component development are, normally, documented in the form of a specification sheet. The integration also happens on different levels. This counts for both the E/E system, the functional integration (validation of the physical properties) and the geometrical integration for the test of geometrical coherence. The sum of all single approvals creates the final vehicle release.

Currently the virtualization in the single disciplines has highly progressed (Fig. 19.6). The typical S-curve by GGI, FGI and SGI shows already a flattening course. However, the progressed virtualization was reached by the application of different IT-systems, at BMW for example more than 1,000 different applications. This leads to enhanced requirements for consistent integration.

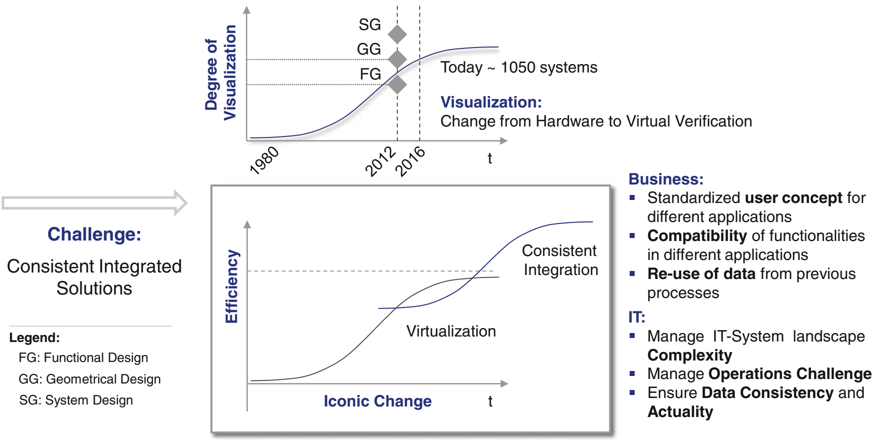


Fig. 19.6 Requirements in the product development

To realize a relevant efficiency increase in handling engineering tasks with additional tools in new application areas is going to be more and more complicate just by increasing the tool coverage. A new level of efficiency can be reached, if once created data are made available for downstream systems. Though, the downstream usage requires consistent integration among processes and used systems.

Basically, different levels of consistent integration can be distinguished (Fig. 19.7). These are the horizontal, the vertical, and the interdisciplinary integration. Additionally consistent integrated system GUIs (frontend integration) are required in the engineering workplace.

Horizontal integration is focusing the process from early requirements management up to the final product release. This can be seen in a time and a procedural sequence. Time: the amount of requirements which should be tested at the appropriate development milestone has to be known (maturity management). Procedural in the term of that the solution elements which belong to the requirements and the appropriate test cases have to be known (traceability). Vertical integration assures that requirements, solution elements and test and release procedures (divided in complete vehicle, subsystems and components) have to be modeled gradually up to a balanced structure. The interdisciplinary integration demands the unique identification of similar assemblies in the different disciplines. Finally the frontend integration limits the number of different user interfaces, the multiplicity of standard products in use and harmonizes the system interaction. The usage complexity has to be reduced, for example through role based user interfaces.

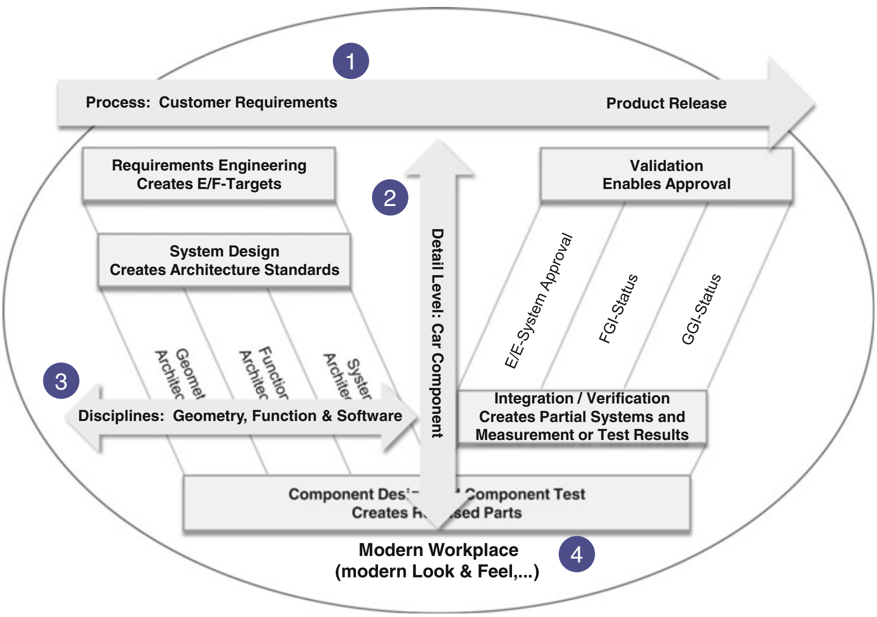


Fig. 19.7 Dimensions of integration in the V model

Fig. 19.8 Change between client and developer view

The horizontal integration has to interact especially with the matrix-like relations between the client view (properties and functions) and the developer view (components and parts) (Fig. 19.8). The combination of the components appropriate to a function or property is named "effect chain". Because components or parts, normally, add something to multiple properties and functions, a matrix-like relation is generated.

Though, relation matrices arise in terms of vertical integration also on subsystem, component or on software module level each with a different level of detail. The requirements (properties and functions) change in the early phase of the development process, too. In addition, the management of a complex product program challenges the structure of the report level, which, normally, align themselves at the organizational structure of the enterprise. Because of this, the management of these always simple looking matrix relations is anything but trivial.

Modern IT-system solutions provide the RFLP-approach to present the necessary matrix structure between client and developer view. Properties and functions can visualized with requirements management systems and function modelers. Thereby, the client view can be modeled. For the developer view methods are required which show logical structures (behavior models, block diagrams for draft models) and physical structures (CAD-models, CAE-models and IT systems to manage the prototype parts) (Fig. 19.9).

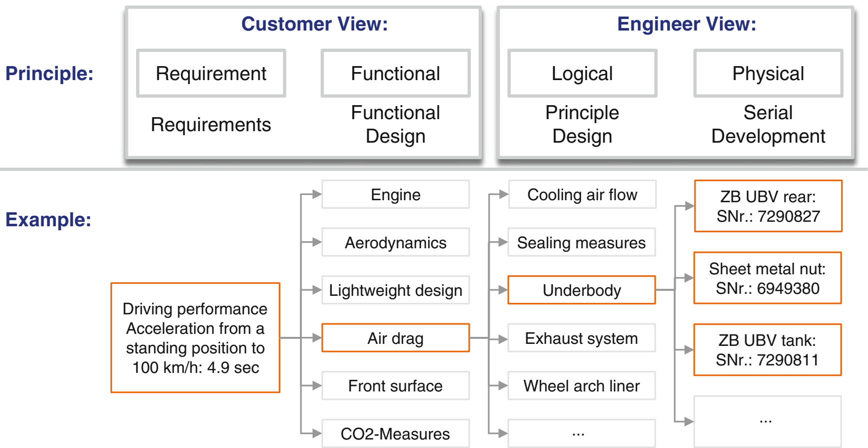


Fig. 19.9 Client and developer view—the RFLP-approach

19.3 System-Level Integration

To meet the introduced demands for integration requires an adjustment of the system scenery which is used in the product design process, especially, concerning functionality and interfaces. Figure 19.10 shows again the dimensions of integration. In this figure is also a classification of the requirements and the solution description or rather the coverage (test cases) in the V model done. Each of these as pyramids illustrated disciplines can be divided in the elements mechanic, mechatronic and software. It is important to know, that the discipline integration is not sequential and uni-directional. Instead a bi-directional and continuous integration scenario needs to be implemented. This is shown in Fig. 19.10 by a sinuous line. This interaction scenario happens on every level between all engaged disciplines.

Today, targets, requirements, functional aspects, logical connections and, finally, the geometry are managed in a heterogeneous system landscape. The demands of integrating this information is not new—but, is often met with proprietary solutions, single interfaces or by using excel spreadsheets. This approach isn't sufficient anymore for the requirements addressed in Fig. 19.11 on product and function-diversity.

General requirements have to be met before starting the system level realization, such as unified ordinal structures which are organizing all development data. These structures are ideally aligned to end customer functions. Furthermore, in the system design process the integration aspect needs to be considered. This can happen for instance through the use of standardized data models and the availability of interfaces [see code of PLM openness (CPO)]. For example it has to be possible to find

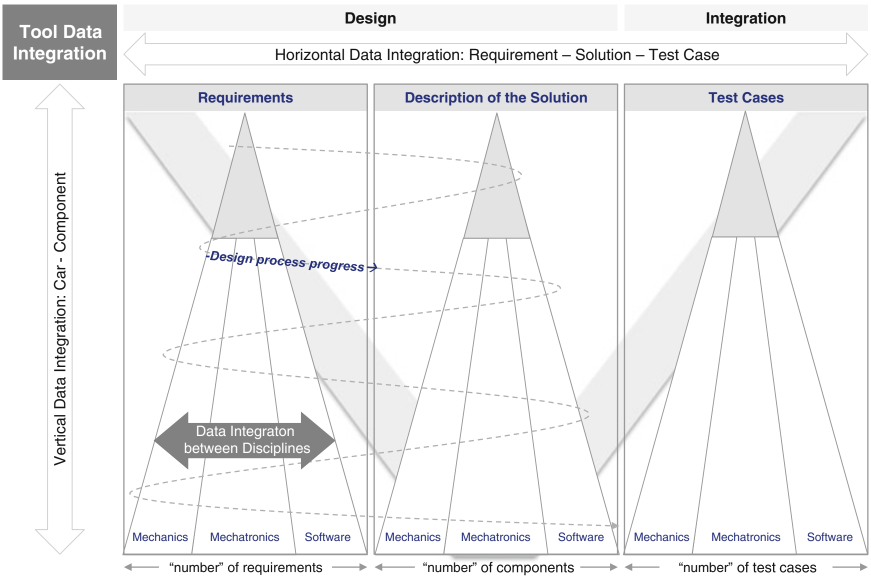


Fig. 19.10 The dimensions of consistent integration

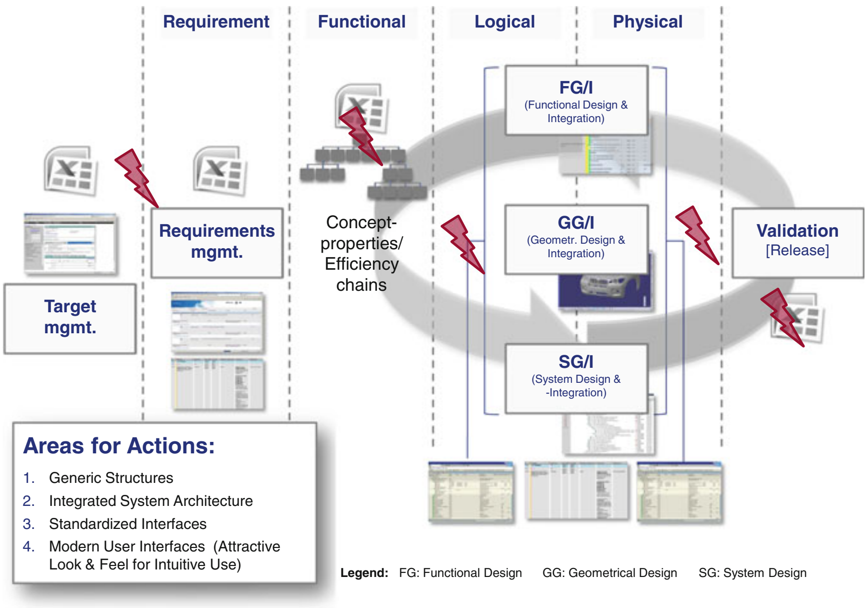


Fig. 19.11 Consistent integration: requirements

for a given validation case the relevant part versions. It has to be also possible to reach for every part or part version the requirements based on which the part has been developed.

19.4 Frontend Integration (Harmonized User Interfaces)

One possibility to realize these requirements are integrated/harmonized graphical user interfaces (see Fig. 19.12). Today, BMW uses a multiplicity of systems in the product development process. Each of these systems has its own graphical interface, often with complete different logical and graphical display options. The ordinary user, who works with up to 30 different systems, is exposed to an unacceptable complexity.

Because of that, different BMW projects address the design of an integrated work environment. The engineer does not need to be aware that the visualized data are originated from different systems, because he always works with an interface that consists mainly of the following components:

- "Google-like" search with a browser like interface to search over different data sources (for example PDM/TDM systems, requirements management, quality and maintenance systems). The browser also integrates the eBOM and product structure management as well as visualization tools.

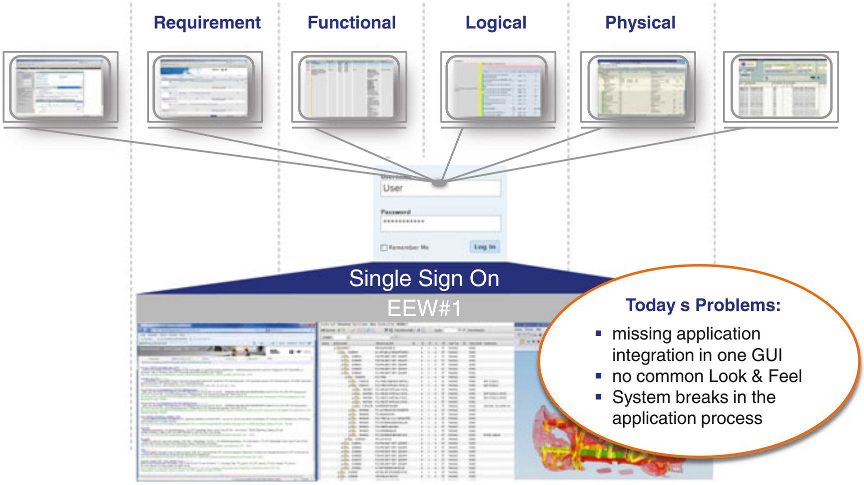


Fig. 19.12 Consistent user-interface

- eBOM and product structure navigator
- Visualization tool, to visualize the products and components in every stage of development. The visualization tool is going to be used also as a reporting tool (Visual Reporting) for the presentation of different parameters (weight, level of maturity, quality, guarantee incidents).

19.5 Consistent Integration in System Design

To understand the problem in the context of the whole product development process, it is necessary (Fig. 19.13):

- To align the process architecture map at the V model to harmonize cross discipline processes. Communication takes place over clearly defined working structures. The integration of results happens at synchronization points.
- To measure the target system architecture with the integration criteria: the relevant RFLP data have to be at least exchangeable over consistent interfaces or better already to be managed in a homogeneous infrastructure (TDM/PDM backbone). A consistent product structure has to be established.
- The concept of integration enables the consolidation of processes, systems and data with the objective of the availability of highly cross-linked development data over the product development process (Fig. 19.14). This integration has to happen both on the organizational level (clear communication rules by the use of V model aligned processes) as well as on the technical level (consistent interfaces and data models). To achieve the integration potential on system level

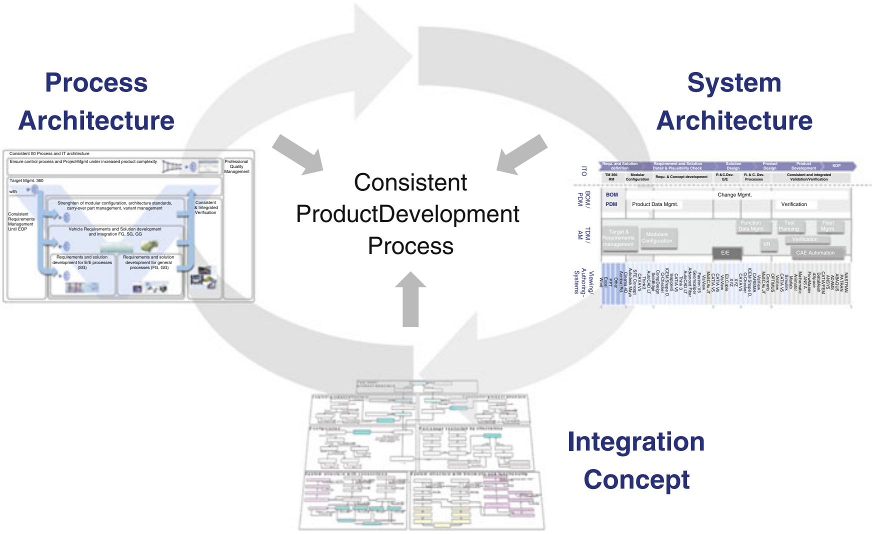


Fig. 19.13 Corporate strategy of the ITO architecture

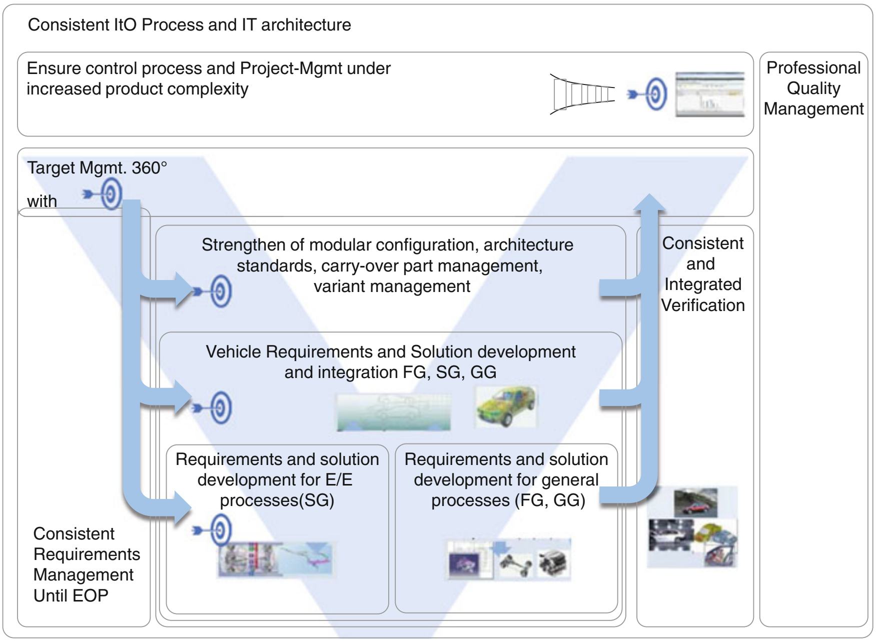


Fig. 19.14 Simplified example for the target process map

the "CPO" has taken a major role to improve the constructive collaboration between users and IT systems vendors since 2012.

Figure 19.15 shows BMW's simplified present-day infrastructure in the ITO process (Idea to Offer, synonym for the product development process). The homogeneous architecture on the BOM/PDM level follows a very heterogeneous infrastructure on the team data management (TDM)—level. Approx. 1,000 different authoring systems are presently managed by approx. 40 TDM—Systems. The TDM architecture is very heterogeneous and incomplete, often only excel- or file-based implemented.

On one hand, the multiplicity of authoring systems is necessary to sustain the access to new technologies and developments. On the other hand, redundancy has to be avoided to reduce IT architecture complexity. The location of the applications respective to the implemented functionalities in the V model helps to identify and to fix these redundancies with adequate IT architecture measures.

The administration of authoring systems, especially the created product data, needs a re-organization with the objective of a significant reduction of the number of used systems, to reach the introduced consistent integration objective.

Figure 19.16 shows the vision of such a modification on the TDM level. Basic functionalities (geometry data-management, visualization, workflow) are realized with a TDM backbone. Based on the TDM backbone enhanced functionalities like

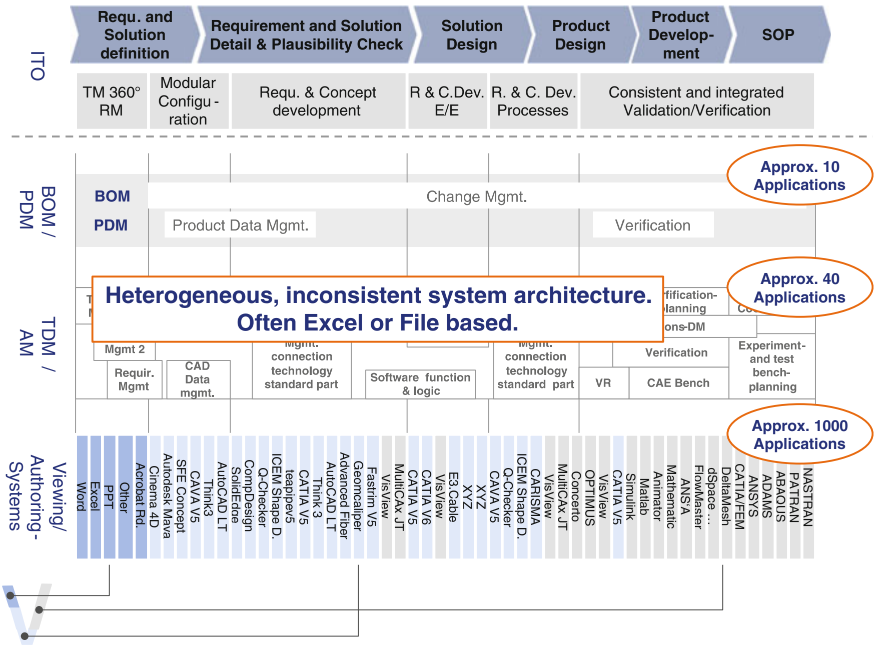


Fig. 19.15 Systems in the product development process (example BMW)

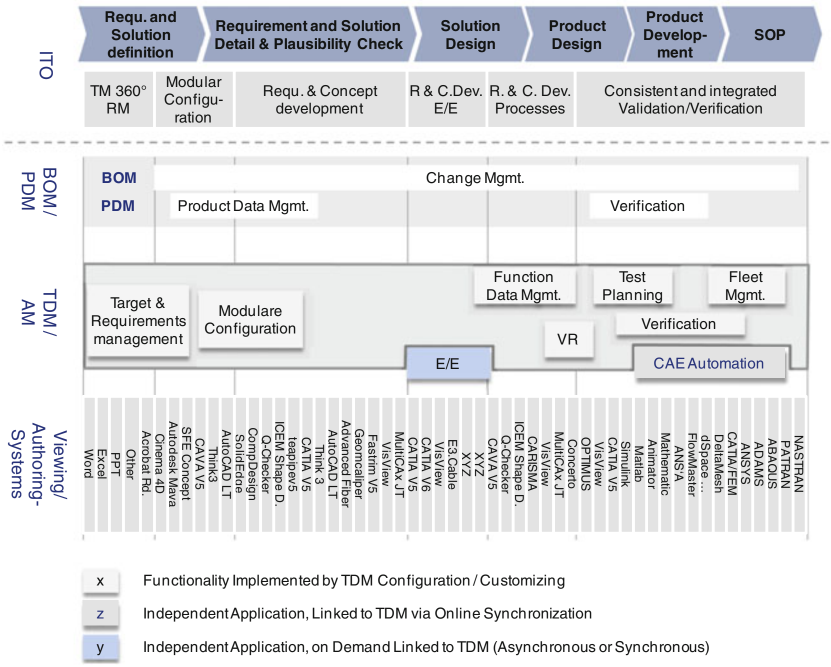


Fig. 19.16 PDP systems (example BMW)—vision

requirements management, modular product structure, function and test data management can be configured.

Because of the multiple requirements it does not make sense to implement all TDM functionalities with one single Backbone. Additional TDM applications might also exist. However, these have to be linked synchronously or asynchronously—appropriate to process requirements.

For the realization of such a consistent integration and the systems engineering approach in the company itself, the commitment of the IT system vendors with regards to openness and standardized interfaces is a necessary basic requirement.

19.6 Code of PLM Openness—Openness as a Requirement for the Implementation of Systems Engineering in a Company

These requirements can be met by using the CPO. The development of the CPO began in the end of 2011 in a close cooperation of different companies like BMW, Daimler, VW et al. (http://www.prostep.org/de/cpo/unterzeichner.html) plus a wide range of system vendors, managed and funded by the ProSTEP iViP Association,

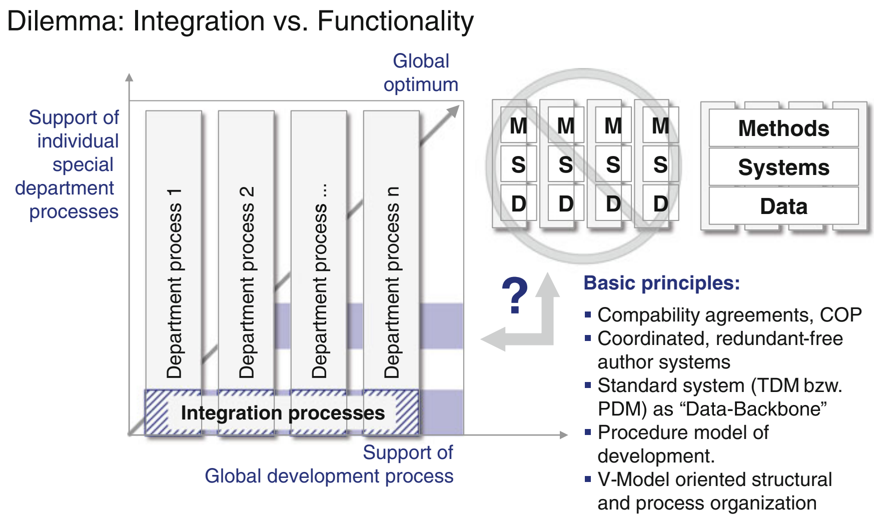


Fig. 19.17 Optimal engineering processes (functionality) versus optimal unified development (integration)

and was released in a first version in the beginning of 2012 (http://www.prostep.org/de/cpo.html). The code defines the term "openness" in the PLM context and defines a basic understanding in the following theme complexes:

- Interoperability and extensibility.
- System architecture and infrastructure.
- Interfaces and standards.
- Partnerships between IT system vendors and their clients

Meanwhile (as of 12/2014), over 80 companies signed the COP and, thereby, made a big step to support the necessary openness.

However, the code is only one element of the complete strategy. The code supports the creation of consistent integration in a complex and heterogeneous system landscape, but cannot be a replacement for an adequate process and system architecture strategy. The dilemma, to optimize on one hand discipline-related processes with highly specialized solutions and on the other hand to integrate all these applications to the perfect optimum, is not solvable with the COP alone. Today, the different methods, systems and data models don't follow a corporate schema. Thus, the incompatibility is quasi implied.

To what extent this applies can be affected with an adequate process-IT architecture strategy, so that in sum, still a good approximation to the targeted optimum can be reached (see Fig. 19.17).

Here, it's crucial to follow these basic rules:

- Harmonization of discipline-specific processes and integration processes with compatibility agreements concerning methods and procedures in the areas of interaction.
- Balanced selection of redundancy-free authoring systems, which are open as defined by CPO.
- Definition of a standard system (TDM or PDM) as Data-Backbone and Workflow-System. Here, the usage of CPO definitions is of significant importance to support also the integration of collaboration processes.
- Usage of a federal approach in the IT Architecture process under strict consideration of central defined rules for integration and compatibility.
- Establishment of a company organization structure that is oriented at the V-Model for both the engineering departments and the IT-department. Only by doing so, continuous solutions can arise and can be operated.

Chapter 20
Concurrent Engineering and Integrated Aircraft Design

Richard Curran, Xiaojia Zhao and Wim J.C. Verhagen

Abstract With the increasing size and complexity of development projects at large companies and organizations in the aviation industry, concurrent engineering (CE) and integrated aircraft design has become of crucial importance in the design process of new products. In order to remain a competitive position and achieve a customer driven approach, aspects of the product's life cycle should be adopted at an early stage in the design process. These aspects include, among others: the overall cost performance and the ability of new system integration. This chapter discusses the implementation of CE in the life cycle of aircraft and systems in general. Challenges related to process parallelization and multidisciplinary design, involving the exchange of knowledge and information throughout the design process, are covered. Supporting techniques along with practical case studies are presented to illustrate the implementation of CE and IAD in real life. Expected future developments with respect to CE as applied to aviation conclude this chapter.

Keywords Aviation · Aircraft life cycle · Concurrent engineering · Design process integration

20.1 Introduction

The civil aviation industry plays a crucial role in fostering trade and making the world quickly accessible and connected. As of 2014, world civil aviation generates a total direct output of $606 billion and is responsible for 8.7 million direct jobs [1]. It has been reported [2] that in 2009 the civil aviation industry in the U.S. provided 10.2 million jobs, contributed $1.3 trillion in total economic activity and accounted for 5.2 % of total U.S. gross domestic product (GDP); these estimates clearly

R. Curran (✉) · X. Zhao · W.J.C. Verhagen
Faculty of Aerospace Engineering, Air Transport and Operations,
Delft University of Technology, 2629 HS Delft, The Netherlands
e-mail: R.Curran@tudelft.nl

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_20

incorporate indirect economic output. Air carriers transported 3.1 billion passengers in 2013 as well as a mere 0.5 % of world cargo volume—which however accounts for over 35 % of world cargo value. In 2012, commercial aircraft production was shown to be in a prolonged up-cycle, shown in Fig. 20.1 [3, 4] and largely driven by the growth of passenger travel demand in Asia and the Middle East. Moreover, the innovation in aerospace technology, such as the new engine development for the Airbus 320NEO and Boeing 737MAX, also generate significant product demand. It is forecasted that between 29,226 and 35,280 commercial aircraft are expected to be produced over next 20 years [5, 6], with estimates recently revised upward [7].

Commercial aircraft can be subdivided into a range of products. Typically, seating capacity, configuration and range are taken as the primary characteristics to segment the aircraft market. Starting from the small, the business jet aircraft segment serves the need for personalized transport; business jets are typically employed to transport small groups of people from point-to-point. The main manufacturers in this segment are Bombardier Aerospace, Gulfstream Aerospace, Dassault, Cessna and Embraer. The regional airliner segment of the market serves capacities between 20 and 100 passengers on short- to medium-range flights, typically for continental routes that act as feeder routes (or 'short-hops') in the conventional hub-and-spoke system of airline transport. The regional airliner segment can be further characterized by considering the two main types of aircraft: turbofan-powered and turboprop-powered. Turboprop aircraft have the longest history in the regional market and typically have greater fuel economy [and thus lower direct operating cost (DOC)] and lower noise when compared to turbofan aircraft. However, the latter can operate at higher cruising speeds, which can lead to higher utilization and consequently higher operating revenue. In practice, both types are utilized in the regional airliner market. Major manufacturers operating in this segment are Embraer, Bombardier, and ATR.

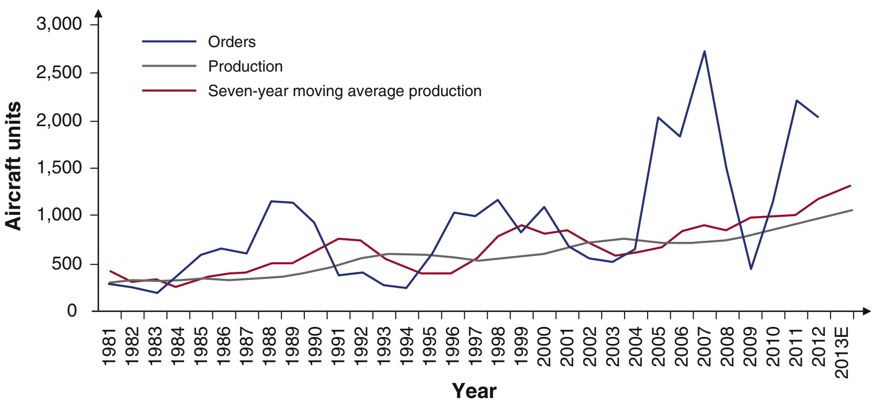


Fig. 20.1 History and forecast for large commercial aircraft orders and production (1981–2013) [3, 4]

Perhaps the most recognizable aircraft market segments are formed by narrow-body (single-aisle) and wide-body (multi-aisle) transport aircraft, having a trans-continental range. In this market, Boeing and Airbus are the primary manufacturers, each offering a range of aircraft to serve a wide variety of routes and capacities. Though much is made of the distinction between hub-and-spoke and point-to-point airline strategies and associated developments in manufacturer portfolios, both major OEMs offer a range of products that can fit within both strategies. In recent years, Embraer, Bombardier and COMAC have been emerging as entrants in the narrow-body market, as they respectively offer the E195, CSeries and C919 narrow-body aircraft. These aircraft are positioned to compete with the workhorses of the Airbus and Boeing aircraft families (the Airbus A320 and Boeing 737).

With the reduced armed conflict in Afghanistan and Iraq, and a reduction in budget for traditional military active governments, global defence spending has declined. The impact of this downward trend is partly attenuated by an increase in defence spending in other countries such as the Middle East, India, China, Russia, South Korea and Brazil. Nevertheless a downward trend can be observed of global revenues for defense companies, which declined 1.3 % in 2012 and 1.9 % in 2011 (Fig. 20.2) [4].

In order to cope with this changing environment, the global defence industry has to find a way to grow profitably in a declining market and maintain an acceptable financial performance by reducing their costs.

In contrast, as highlighted, significant growth is forecasted for all segments of the civil aviation market. This puts significant requirements on its main stake-holders, including original equipment manufacturers (OEMs), aircraft operators and civil aviation authorities. It demands not only highly efficient production process with a large production capacity, but also user-friendly and environmentally friendly design, manufacturing and operation. This can be evaluated by certain

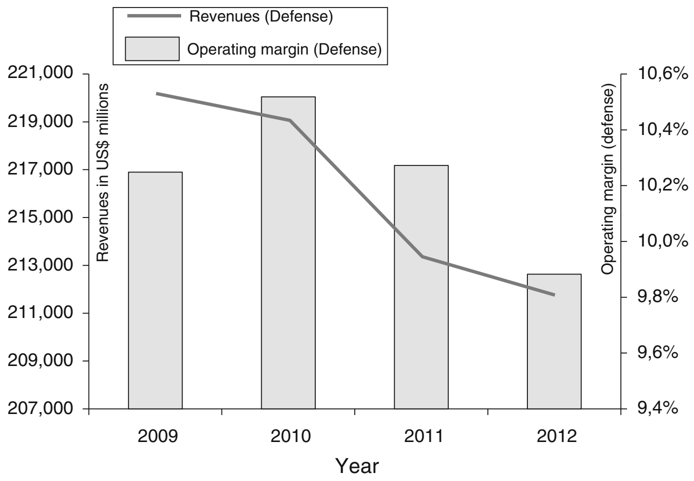


Fig. 20.2 2014 global aerospace and defense industry outlook [4]

performance indicators during the aircraft life time right through from concept design to the ultimate disposal, such as safety, economics (low cost), comfort (good infrastructure), noise (less noise), cleanliness (less emissions) and energy efficiency (less fuel burn). They are generally regarded as life cycle performance indicators and along with market expansion, are seen to represent the critical performance criteria of the aviation industry [8].

One of the main issues of the aviation industry that could limit market growth is the environmental impact caused by air transportation. With the expected three-fold air travel over the next 30 years, environmental awareness has become even more important [9]. With a current yearly production of 628,000,000 tons of CO_2, which represents 2 % of the human induced CO_2 emissions, the aviation industry has to react on this further evolving threat [10]. In Fig. 20.3, a prediction of the annual growth of international aviation emissions, made by ICAO [11], is given.

From this graph it can be seen that the global air transportation induced emissions will increase by a factor of five in 45 years time, if not reacted upon adequately. Multiple initiatives, such as Europe's Advisory Council for Aviation Research and Innovation in Europe (ACARE) [12], have been started to reduce environmental impact, which besides CO_2 also consists of NO_x emissions, perceived noise, and the environmental impact caused by aircraft manufacturing, maintenance and disposal [10]. In order to anticipate on this, companies involved in the aviation industry should strive for improvement of the efficiency of aircraft and engines, and improve the aircraft lifecycle and current Air Traffic Management system.

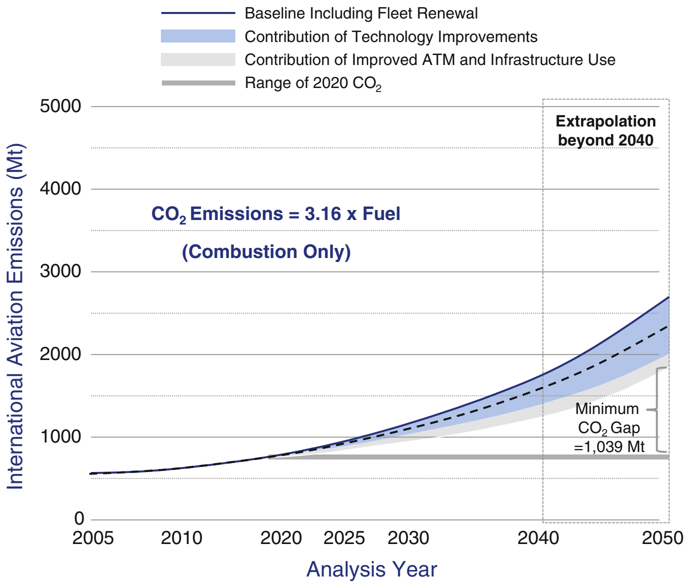


Fig. 20.3 ICAO/CAEP fuel burn trends from international aviation, 2005–2050 [11]

These developments highlight the need for an integrated and advanced design process that is able to ensure the concurrent synthesis of many life cycle performance drivers within a complex and collaborative aviation enterprise. It is the aim of this chapter to present recent developments in aviation research that contribute to this overall need.

The structure of this chapter reflects this focus. In Sect. 20.2, the aircraft life cycle including its phases and components are discussed. Subsequently, the aircraft design process is described in Sect. 20.3, which is then put into the context of concurrent engineering (CE) and its application to aviation in Sect. 20.4. In Sect. 20.5, applications of various elements of CE as presented in this book are discussed; this includes the areas of Multidisciplinary Design and Optimization (MDO), digital mock-up (DMU), value engineering (VE) and life cycle costing (LCC) within the context of aviation. Finally, a concluding section gives insight into future research and development of aviation from a CE perspective.

20.2 The Aircraft Life Cycle

The civil aviation activities relevant to the aircraft life cycle are categorized in Fig. 20.4. For each phase within the aircraft life cycle, a process or activity series are identified, along with the associated participants and relevant entities.

At the early phase of the life cycle, the research and development phase starts with identifying the current market needs. Standards and design requirements are then established so that based on a list of requirements (LOR), designers can generate promising aircraft concepts, accompanied by a series of feasibility and

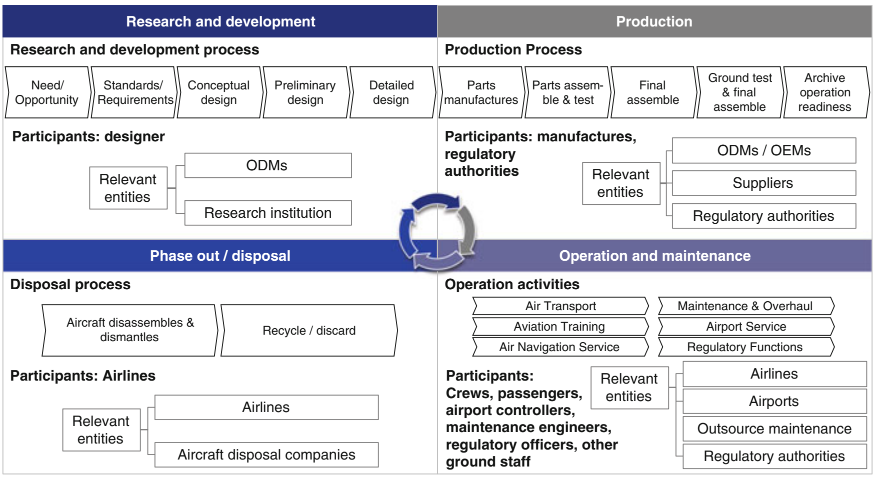


Fig. 20.4 Aviation industry relevant to aircraft life cycle [13–15]

verification studies. In practice, the design of aircraft is organized into design programs: a dedicated organization-within-an-organization aiming to design a (family of) aircraft and the supporting activities throughout the lifecycle (i.e. not only the product is designed, also production, operation and support are developed). For instance, as a global aircraft manufacturer, Airbus has design and engineering teams at multiple sites around the world that are involved in the same aircraft program [16]. In order to ensure that the required knowledge is available throughout the design process for each department, Airbus's headquarters in Toulouse, France gathers the top-level competencies. These include the architecture integration, general design, structural design and computation, integration tests and systems, and propulsion.

The actual design exercise starts from the conceptual design phase according to the performance and life cycle goals identified, and the designers are required to generate possible competing concepts, after which iterations on performance evaluation and optimization (on an aircraft level) are performed, leading to the selection of a baseline configuration. The output of the conceptual design is a 3-D geometric representation of the baseline aircraft design with associated performance indices. Subsequently, the concept is further developed through the use of parametric sizing studies in the preliminary design phase. The size of the baseline concept is refined while the aircraft level configuration is frozen, while modest changes on the sub-assembly and component level are still possible. The main deliverable of the preliminary design phase is a 3-D drawing and representation of the aircraft concept with sized components. Finally, the detailed design phase involves the precise design iterations from the global level for the whole aircraft to the system design level and ultimately the local level associated with detailed parts design. The final outputs are the detailed production drawings, finalized aircraft specifications and performance properties. Furthermore, design of the production process is carried out concurrently during the preliminary and detailed design phases. The entities involved in the research and development phase are the design group from the original design manufactures (ODMs) or the OEMs (such as the Airbus and Boeing companies) and research institutions such as aerospace research laboratories (e.g. ENAC, DLR and NLR) and of course universities.

Manufacturers and designers are working closely during the production process. Parts manufacturing is initiated firstly, followed by sub-assemblies for manufactured parts and components, and then the final assembly process is carried out. The testing of components and systems are conducted during the whole phase and once the ground tests are completed multiple prototypes are prepared for the first flight and a series of subsequent flight tests. When the aircraft is validated to have achieved all the standard specifications an airworthiness certificate can be issued by the regulatory authorities. Then mass production is initiated based on the orders received, with Airbus for instance building more than 1 A320 aircraft per day as per 2009. In addition, the aircraft needs to achieve operational readiness. The OEMs and ODMs invest significantly in intensive and automated manufacturing capabilities for the whole production and assembly process. Other supply chain entities and outsourcing manufacturing companies play a major role in the extended enterprise and the OEM becomes more of a designer and integrator, and there is additional input to be integrated from regulatory authorities such as FAA, CAA, EASA.

Operation and maintenance activities define the life cycle once the aircraft enters into service. Associated activities include air transportation operations, aviation training, air navigation service provision, maintenance and overhaul services, airport services and regulatory functions. The activities of the operation and maintenance process are performed simultaneously and recursively. As aircraft age, 'heavier' maintenance checks and overhaul activities are scheduled to keep the aircraft in an airworthy state. Other stakeholders during this lifecycle phase of course include the passengers, airline crews and tickets staff, air traffic controllers responsible for flight and ground control, and again regulatory authorities for example responsible for the continued airworthiness of each aircraft.

Aging aircraft are retired, sold-on, or disposed of according to the airlines' fleet management strategy. Based on the aircrafts' service condition, the disposal process is defined, and normally involves being 'parked' in a dry desert graveyard or being disassembled so that the dismantled parts can be recycled or sold-on by outsourcing companies involved in end-of-life-solutions.

Cost performance is concerned with every aspect of the life cycle. It is the fundamental driving element within the aviation industry and air transport market, along with safety. All associated and relevant industry activities are assessed and even enabled by effective cost performance. From a depth perspective, all of the relevant disciplines and parameters for an aircraft or within air transport are highly interrelated and mutually influential, including for examples aerodynamics, materials, structures, systems (such as avionics, hydraulics and power), cost, market demand, environment impact and energy utilisation.

20.3 Integrated Aircraft Design Process

Considering the aircraft life cycle, the vital factor which controls the final decision of a bid is always cost. The cost performance needs to be evaluated within every aspect of the life cycle, including the life cycle activities and all participants. The cost associated with each phase of the design is shown in Fig. 20.5. The reduction of cost is always the goal of the whole aviation industry, along with extremely safe operational performance. Consequently, the evaluation of cost in an accurate and effective manner is always the goal of the analyst. While for cost engineers, it is important to link the design and cost properties together, and to reduce the cost while keeping the aircraft at a required performance and technical operational level. This demands an integrated design and development process, in which life cycle performance and requirements are considered at the early design stage.

After a century of design practice, the integration of disciplines and design process has evolved continuously and design activities have changed from a specialization focused approach to a more systems focused approach already in the

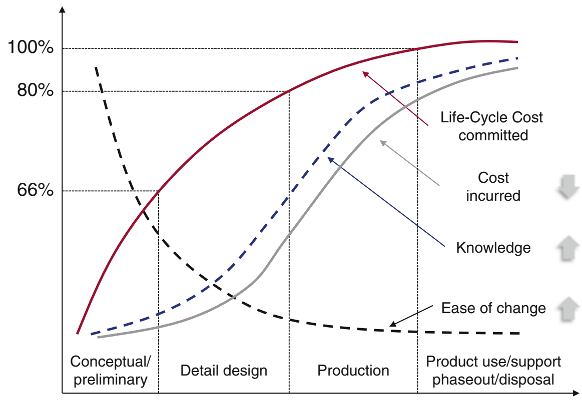


Fig. 20.5 Aircraft life cycle and cost, knowledge relation [13]

1950s. However, the analytical specialist remained more influential in some ways than the design engineer. In the 1970s, computer-aided design (CAD) exploded on the scene along with the promotion of a life cycle cost (LCC) approach within the design process, with the balance between performance, LCC, reliability, maintainability and safety being facilitated through the emergence of advanced information and computing technologies in the 1980s [17]. The trend is always on improving the design capability for reducing development time, and achieving more complete design synthesis at an earlier time.

Various advanced methodologies and technologies have been embraced over the years by aviation in order to advance the integrated design and development process, including: CE, product life management (PLM), multidisciplinary design optimization (MDO), DMU, collaborative engineering (CE), Digital Manufacturing, knowledge based engineering (KBE), etc. Each has definite strengths and is also inter-connected with the CE philosophy that has been developed initially in the 80s. This emphasizes the need for concurrent design and analysis that incorporates all aspects of the aircraft life cycle, integrating their influence into the design decision process and also helping to make the process more efficient. Gradually, based on the CE principle of process parallelization, the combined and integrated analyses and optimizations on multiple disciplines was promoted and facilitated. Ultimately, MDO in its broadest sense addresses the integration of aerospace analytical disciplines such as aerodynamics, propulsion, structures, and control—as well as manufacturing, operations and maintenance issues in the life cycle context. By employing mathematical optimization methods, a minimum weight or cost design can be achieved [18]. It can be used to strengthen the conceptual design process by providing more analytical design space for multi and inter-disciplinary integration and ultimate optimization. MDO has been applied successfully in multiple design programs, for instance the design of the Airbus A380, where numerical structural optimization incorporating lifecycle constraints [19] has resulted in significant weight savings. Multiple other success stories are available for the aviation domain—see e.g. Chap. 15 and Sect. 20.5. In addition, KBE techniques can now be employed to link the development of the central design geometry with the necessary extensive supporting knowledge so as to improve the efficiency of performing often repetitive and time consuming tasks, which frees the designer and engineers up for focusing on innovation and creative solutions [20]. Examples for the aviation industry are discussed in Chap. 10. A recent development is embodied in the value driven design (VDD) approach which embraces the concept of MDO but promotes it in a more performance driven way. Such VE techniques can be adopted in order to produce a balanced measure of product function, cost and ultimate utility. More theoretical background and practical examples of VDD are given in Sect. 20.5.

20.4 Concurrent Engineering Within Aerospace

CE was a term first coined by Winner et al. [21] of the DOD Institute of Defense Analysis and is defined in full in Chap. 2. The definition stresses the parallel, concurrent, execution of product and process design activities by integrating multiple design disciplines and upstream and downstream functions involved in the lifecycle of a product. CE is known under various names such as Simultaneous Engineering, Concurrent Product Development, and Integrated Product Development [22–24]. It has been noted that there are three fundamental characteristics: the early involvement of key participants, the team approach, and the simultaneous effort on different phases of the product development [25]. CE teams typically consist of the functions marketing, product engineering, process engineering, manufacturing planning, and sourcing activities. The principle focus initially was on the integration and alignment of design and manufacturing functions, while taking into account consumer demands and supplier capabilities.

Cross-functional CE teams incorporate experts focused on different aspects such as marketing on usability, engineering on functionality, production on manufacturability, and purchasing on affordability [26]. In such situations, communication needs to be predominantly personal and involve face-to-face contact [27]. The early involvement of relevant stakeholders in the design and development process enables exchange of preliminary information, thereby potentially reducing the number of engineering change orders, which are often the reason for delay in product development projects. Strategies for the exchange of preliminary information exchange may differ with the level of downstream uncertainty and costs of process idleness [28].

In order to support collaboration in teams and facilitate information exchange and use, significant effort has been made to develop engineering knowledge and collaboration tools [29], although these are still limited [30]. Lu et al. [30] with reference to the VIVACE European project [31] has reported that 26 % of project meetings in Airbus involve international partners and more than 400 one-day trips were taken by Airbus engineers to collaborate with other project members on a daily basis. They also spend an average of 49 % of their daily activities in meetings

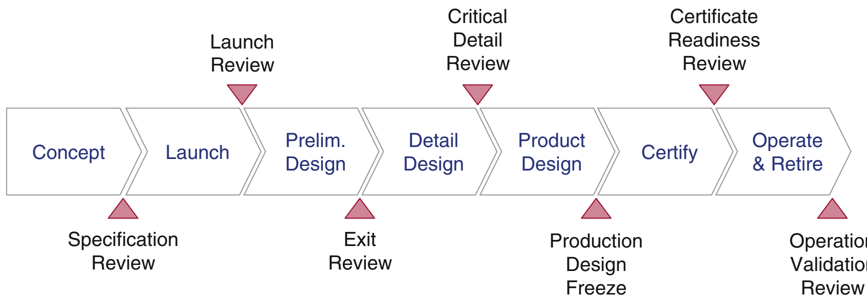


Fig. 20.6 The aircraft life cycle process

and discussions with stakeholders. In addition, paper documents and electronic files, like e-mail records, are still the standard for supporting these meetings, while 50–80 % of the documents are in paper form and 70 % are for multi-cultural working sessions only.

The life cycle process within aerospace is conceptualised in Fig. 20.6 [32], showing the major phases in the life cycle. This illustrates the challenge of a more serial view on the management of information throughout the life cycle, where ideally, any information and analysis relevant to the concept stage from the subsequent stages is available during the conceptual design stage; like certain regulations regarding retirement, which may already be considered within the conceptual stage.

Based on the fundamental principle of data/knowledge sharing within CE, it can be seen from PLM systems that the integration and the optimal running of tasks may be achieved by establishing a knowledge hub that includes Product, Process and Resource information and forms, or attributes. In addition, another implied key element from PLM is information/knowledge storage, control and utilisation! This all part of the PLM paradigm that is the vision of associated software and framework suppliers such as Dassault Systems, Parametric Technology, and Siemens PLM Software.

The collaborative effort and organisational challenges of the whole CE endeavour is extremely challenging and early attempts at solving this are exemplified by the concurrent design facility at the European Space Agency (ESA), as illustrated in Fig. 20.7. It is interesting to note that the disciplinary experts arranged around the outer space, with access to their tools through the desk-top workstations, and that their input is then facilitated through the concept of a multimedia wall that primarily helps to provide diverse and fragmented information in an effective manner to the whole team [33]. In relation to this, a methodology to visualise aircraft design tasks is further explored in Dineva et al. [34].

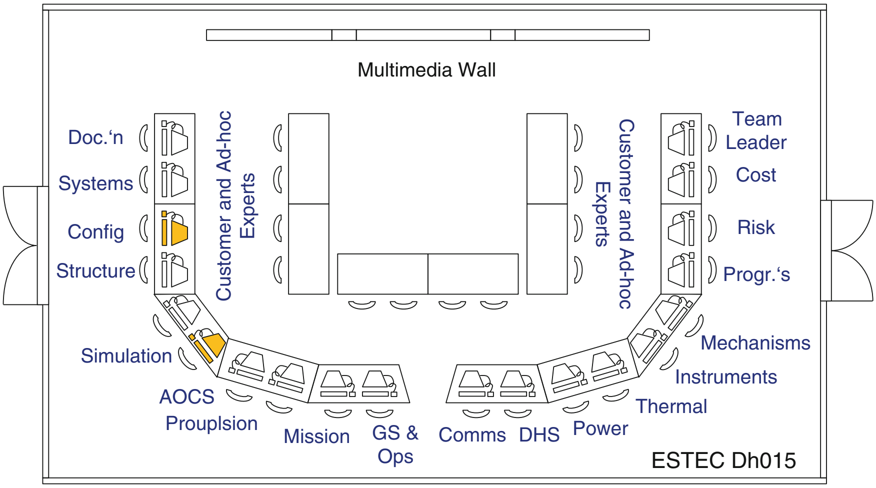


Fig. 20.7 The integrated multidisciplinary design facility at the Europe Space Agency

20.5 CE in Aviation: Supporting Techniques and Use Cases

As part of recent developments in the application of CE to the aviation industry, a number of supporting techniques, associated brief theoretical background and examples of application to the aviation domain are highlighted in this Section, with a particular emphasis on application. The following techniques and methods are discussed: CE, DMU, MDO, VE, LCC, and Systems Engineering (SE).

20.5.1 Collaborative Engineering

The desire for incorporating multiple lifecycle considerations requires tight integration of multi-disciplinary knowledge and collaboration between engineers across various cultural, disciplinary, geographic and temporal boundaries [35], whereas environmental concerns have also added to product design and development complexity [36]. As discussed at more length in Chap. 2, putting the emphasis on collaboration has led to the term Collaborative Engineering (CE*) with the following definition [37]:

> Collaborative Engineering is a systematic approach to control lifecycle cost, product quality, and time to market during product development by concurrently developing products and their related processes with response to customer expectations, where decision making ensures input and evaluation by all lifecycle functions and disciplines, including suppliers, and information technology is applied to support information exchange where necessary.

In aviation, all these elements are highly relevant; the global supply networks that are currently in use to finance and execute development, production and delivery of wide-body aircraft such as the Airbus A380 and A350XWB and Boeing 747-8 and 787 drive the need for application of CE*. For instance, in a well-known example, the Boeing 787 has been developed and is currently manufactured in a network including dozens of major partners, covering some major continents [e.g. Boeing (USA), Alenia Aerospatiale (Italy), Kawasaki Heavy Industries (Japan)]. Each of these major partners in turn manages their own supply network (with input from Boeing), creating a tiered supply chain network.

Both aviation research and practice have a long-standing interest in lifecycle considerations including cost (see also Sect. 20.5.5), early supplier involvement (ESI) and information (technology) support [38]. In this Section three use cases illustrate the crucial importance of engineering collaboration in the aviation industry. In the first, supply chain harmonization is covered (entailing aspects of ESI and information technology support) by considering the development of the Boost Aerospace digital hub. In the second case, supplier integration and technology support (through PDM and PLM systems) is described for the case of a manufacturer of fixture equipment. Finally, supply chain communication and collaboration for the case of buyer-furnished equipment (BFE) is discussed.

20.5.1.1 Use Case: Boost Aerospace

The long lifecycle of an aircraft requires sophisticated configuration management tools. The aviation industry has major potential in harmonization of its supply chains. To strengthen European aerospace programmes (i.e., product development projects), competitiveness has to be improved at the extended enterprise level. In order to enable and accelerate the deployment of digital processes and tools across the extended enterprise from the OEM to the tiered suppliers and to customers, harmonized solutions and open standards are a key factor for success [39]. The verticalisation of the supply chain requires comprehensive digital PLM collaborative platforms. This requirement was accepted by five leading European aerospace and defense companies (EADS/Airbus, Dassault Aviation, Safran and Thales) which have created a European digital hub called BoostAeroSpace (see Fig. 6.7) for the management of collaborative programmes and their supply chains [40]. It provides highly value-added standardised and secured collaborative services for stakeholders in the entire supply chain. Therefore, these services dramatically reduce the specific environments dedicated to each customer, providing interoperability with their information systems.

BoostAeroSpace provides the following service levels:

- AirCollab (collaborative workspace, e-meetings)
- AirDesign (Product Lifecycle Management (PLM) collaboration, DMU sharing)
- AirSupply (Supply Chain Management (SCM) collaboration, logistics exchanges, vendor managed inventory)

These services have become productive in 2011 and as of 2014 serve more than 300 companies. The platform is used by its founders and their international partners and suppliers. Two main benefits are targeted: first, the use of these standardized services by the main European OEMs is anticipated to dramatically improve the generic collaboration with suppliers and interoperability with their information systems. Second, the platform is to reduce process cycles and overall costs. The mentioned services are provided as "Software as a Service" (SaaS) for all OEMs, suppliers and small companies along the whole supply chain, enabling them to potentially make the same gains in competitiveness as the five founders.

AirCollab provides generic collaboration services based on its customized standard collaborative solution Microsoft Sharepoint. It enables "turnkey" collaboration with external partners and internal teams by using collaboration utilities like e-meeting and pre-defined templates for collaborative project management and information sharing. For the aftermarket it maintains a reference document library.

AirDesign is focused on aircraft program design and manufacturing processes and deploys the Enovia/CATIA V6 collaboration suite of Dassault Systemes. It serves the following five use cases which are typical for almost each collaborative project:

1. Technical data package exchange: Secured data exchange management between partners/suppliers.
2. PLM collaboration using data exchange: Shared product structure based on STEP AP2013 (see Chap. 6) integrating partners/suppliers's product design data deliveries through secured data exchanges mechanisms.
3. Co-review: Allows design co-review on shared product structure between partners connected to PLM hub (enable context deliveries and assembly/sub-assembly review based on shared DMU according to the project scope, see Chap. 13).
4. Share catalog and new part request process: Publication of harmonized standard parts catalogue to be used by partners/suppliers (see Sect. 14.5).
5. PLM collaboration @Hub: Provide collaboration workspaces with generic V6 PLM functionalities (see Chap. 16).

AirSupply is a central aerospace SCM platform that facilitates secured and traceable communication across companies and provides valuable assistance at both operational and management level. As a result, processes with external partners are more transparent and dependable while various alert mechanisms allow exception-based management of the supply chain. It is based on technology from SupplyOn, a specialist in cross-company supply chain collaboration which is already established in the automotive industry. In close cooperation with BoostAeroSpace, SupplyOn's platform has been adapted to meet all requirements specific to the aerospace industry.

AirSupply comprises the following six functions:

1. Demand forecast: Send demand forecast to supplier based on flexible horizon, projected horizon

2. Purchase order: Send purchase order to supplier based on firm horizon
3. Consigned Vendor Managed Inventory: With or without consignment stock, associated to Self-billing
4. Dispatch advice and Receipt advice: Supplier sending dispatch advice and customer sending receipt advice
5. Self billing receipt advice: Customer sending billing to supplier
6. Cockpit and exception: Indicators, alert and exception management.

20.5.1.2 Use Case: Collaboration on Fixture Equipment [41]

This use case reflects the design and manufacturing processes of fixture equipment for the aeronautic industry. The equipment supplier is a basic manufacturer of the assembly tools with sequential design and manufacturing processes. As shown in Fig. 20.8, three departments are engaged in the global process of assembly tools purchasing: production service specifies the assembly needs, the tooling R&D designs the tooling structure, and the purchase service negotiates and sends the order to supplier which are distributed globally [42]. After the completion of the tool, it is sent directly to the production shop for use. In case of changes in design, these modifications imply changes on the specification of the assembly process and, thus, of the assembly tool. The whole cycle of the assembly tool ordering is then repeated to cope with the new specifications. Thus, a new PLM-based approach is developed for the seamlessly integration of all the information specified throughout all phases of the equipment's life cycle between OEM and a new global supplier

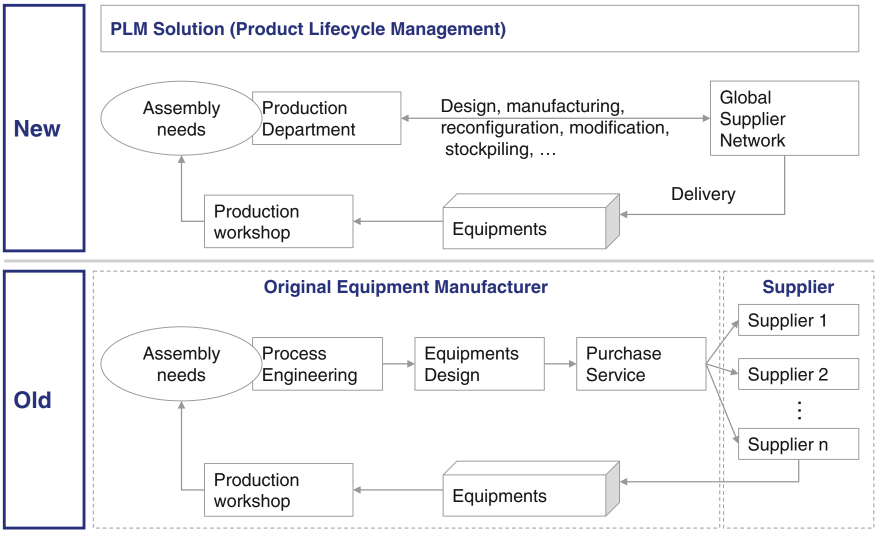


Fig. 20.8 Solution concept for collaboration OEM—supplier

network (GSN). Figure 20.8 also shows the new configuration where tasks of design, configuration and fabrication of the assembly tool are performed collaboratively with the new GSN. Suppliers are simultaneously informed about new modifications of the assembly operations and design the new tool by themselves. The implementation is based on the concept of working situation to describe different relations between supplier network (and assembly tool) and OEM (and aircraft part).

This approach aims not only for better integration of the supplier in the design and manufacturing processes, but induces a new collaboration strategy between OEM and supplier. Suppliers are going to be involved early in each new project. Important evolutions in the current configuration of the development process occur by the shift from a linear and sequential process to a much more "collaborative" one. This reconfiguration leads to significant improvement in saved cost and time in association with a greater innovative potential. The supplier gets a new role, not only as an efficient manufacturer, but more as a partner, collaborating in different product development project stages.

Horizontal collaboration will be improved by means of a "back office" interface that gives suppliers the possibility to share their knowledge and to get a common representation about the project evolutions (see Sect. 7.3.1). Subsequently, the specification and manufacturing of the assembly equipment are performed progressively and jointly by different units of GSN. When the engineering starts the design process of the fixture equipment, production planning might simultaneously schedule the manufacturing operations in order to optimize the production process. Furthermore, during the whole development process, engineering can inform progressively the production and other furniture suppliers about the bill of material structuring the equipment, in order to shorten the purchasing time.

Based on the conceptual specifications, various functions are available by modern, well-customized PDM systems (see Chap. 16) for collaborative work [43]:

Product data interfacing: The OEM defines on his own system the assembly activities and the references of concerned product components. However, to fulfill the equipment development operations, the GSN members should get some information about the OEM product (structure, geometry, materials, etc.). Based on the meta-model structuring the relations between processes and products, the PDM system extracts from the OEM system only the relevant and authorized aircraft data (see Sect. 16.6). In the opposite way, data can be sent for DMU (see Chap. 13).

High level of transparency: The OEM gets more visibility about the supplier's workload and might take into account their constraints when it defines the manufacturing planning. The suppliers get insight in the OEM's planning and project activities very early (see Chap. 7). For instance: when the OEM decides the re-scheduling of its activities, the suppliers are automatically notified by these modifications to ensure their possible reaction. The triggering of the equipment delivery process depends at least on supplying activities that are managed in the OEM organization. It helps to reduce the number of iterations for the cost estimates and negotiation since it is based on common procedures and will be fulfilled through a collaborative process.

Track the project progress: The project coordinator gets more visibility about the GSN workload and the OEM assembly planning. These constraints can be taken into account when managing and scheduling the remainder of the project. For this function, the PDM system extracts planning information from different partner's inputs and aid coordinator to schedule the equipment project.

Collaborative project management: The PDM system plays the role of a mediator between different partners. GSN users download the equipment order with their associated requirements, and upload the different documents defining the corresponding equipment. The PDM system notifies the partners simultaneously by subscription mechanism about the evolutions of both aircraft and equipment projects (see Sect. 16.6.2). At the end of the project, OEM validates the reception of the equipment.

Apart of all benefits, this approach works properly only if a certain level of interoperability is preserved between the PDM backbone at OEM and supplier's IT systems. It includes a high level of subordination at the supplier's side as well as a well-adjusted collaboration model. Like other industries which deal with complex products (automotive, transportation, shipbuilding), this is still subject of basic research and development [44] (see Chap. 6).

20.5.1.3 Use Case: Communication with Buyer-Furnished Equipment (BFE) Suppliers

Like other complex products, airline customers customize a wide variety of airplane features provided by aircraft manufacturers and needed to properly differentiate individual brands and to satisfy operational requirements. Airlines have the choice to modify or add among a wide variety of pre-qualified selections available from a large pool of industry-leading suppliers (see Chap. 14). Options are provided by either Seller-Furnished Equipment (SFE) or BFE. BFE is a term used in the aerospace industry to denote components supplied at no charge to the manufacturer by the purchaser for use in the assembly procured by the purchaser from the manufacturer. Typically, such equipment comprises specific cabin equipment (seats, galleys and galley equipment, entertainment equipment, kitchen, bathroom).

Whilst the SFE supplier is required to be fully integrated in the product creation process of the aircraft manufacturer, there is no strong contractual precondition for similar treatment of a BFE supplier, although it participates in the product creation process of an aircraft. Therefore, several issues in the process chains arise, in particular in data exchange. DMU could become a serious issue (see Sect. 13.3.2).

Basically, there are two possible solutions: use of a neutral process format like JT (see Sect. 11.6) or deployment of a data exchange service portal which supports a plethora of CAD systems and formats (Fig. 20.9) [45]. In both cases sufficient data quality is the decisive impact factor [46] and can be achieved by appropriate methodical measures which include manual rework [45]. As JT is not yet widely

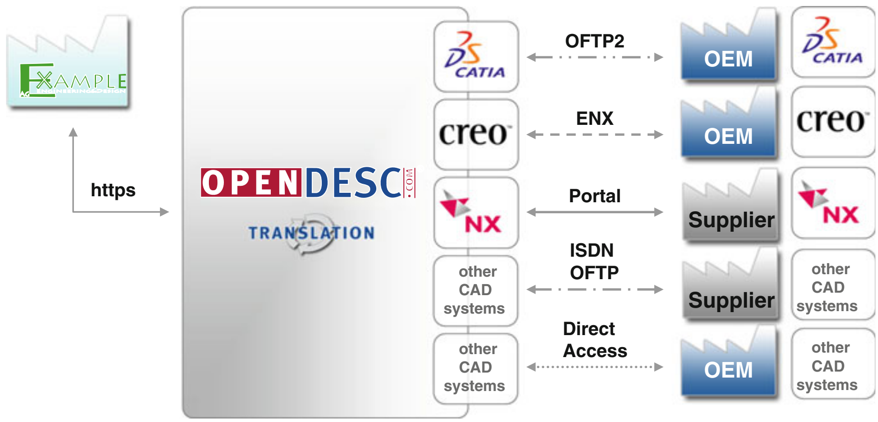


Fig. 20.9 Supplier portal for data translation and exchange [45]

adopted in the aerospace industry, many current programs are conducted using supplier portals which support a multitude of physical connections and transfer protocols too.

20.5.2 Digital Mock-Up

DMU is a core method in CE in the aerospace industry for assembly examination, layout examination, interference checking, and maintainability (see Chap. 13). Based on complete CAD data and a powerful PDM system, DMU can be created synchronously with each design activity. Based on advantages of DMU, the use of the Physical Mock-up has been reduced dramatically over the past years. Beside of standard monitors, many different graphics devices are used for graphical output in aerospace industry (mobile devices, virtual, augmented and mixed reality).

After the complete aircraft has been built in its full-size virtual environment including the adjacent manufacturing or operation equipment, the engineering examination, assessment and decision making can be conducted in a virtual space. The advantages of such DMU are that one can easily establish the specific advantages and disadvantages of any design solution by applying a variety of scenarios to a subject. In addition, during an assembly or decommissioning, any interference or collision with a subject can be elucidated, and possible errors can be prevented in the design. What is more, the location of an operation and the methods can be conveyed to workers by means of a industrialization DMU (IDMU) before an operation, and thus the understanding of an operation could be improved and the time required for an operation could be considerably reduced.

However, the industrial use of DMU in aerospace is struggling with many limitations due to the large scale and complexity of an aircraft. Neither the standard software packages from leading PLM vendors nor the standards viewers nor the communication facilities are able to handle the huge amount of data which is still needed to describe a full DMU of an aircraft. Thus, appropriate examination procedures are needed to perform the DMU tasks in singular zones of an aircraft and, subsequently, to aggregate the results. The DMU for the aircraft systems which are distributed along of entire aircraft, remains as an especially challenging task.

20.5.2.1 Use Case: Final Assembly Line Design [47]

The design of a final assembly line (FAL) at Airbus is carried out as concurrent development process during the product industrialization activity and can be decomposed into three assembly line design phases: concept, definition and development. During the conceptual phase, designers require defining FAL alternatives with different values for the input requirements.

Based on the product configuration and the scenario, Manufacturing Engineering is responsible for executing the case and for defining the DMU of the industrialization solutions or FAL alternatives. Both the scenario and the FAL design are part of the IDMU which comprises product, processes and resources information, both geometrical and technological. At the conceptual phase, the process of generating industrialization solutions depends heavily on personnel experience and is time-consuming. Thus, manufacturing engineers can only check a simplified set of cases to generate early manufacturing processes and resource requirements. In order to enhance this process, it was decided to develop a software application to assist designers in the definition of scenarios and to generate FAL alternatives at the conceptual stage.

A ‘to be’ IDEF0 process model was defined, focused on the Industrialize activity, to conduct the information flow and helps to identify the concepts and knowledge involved in the aircraft FAL conceptual design process. The next step was to develop a knowledge model using UML. The knowledge modeling of aircraft assembly lines requires reviewing works dealing with modeling of assembly information, processes and lines. From this review it was concluded that the semantic concepts involved in the conceptual design phase of an aircraft assembly line were not fully taken into account in the identified models. Models presented in the literature provide three main views: product, process and line balancing. The modeling of the conceptual phase demanded to integrate and to extend concepts from the three views, particularly from the process view. The used conceptual model was divided into three interrelated sections or knowledge units: Product, Processes and Resources, together constituting the IDMU. The product section comprises the concepts to define the joints to be assembled and both the functional (as designed) and the industrial (as planned and as prepared) views. The process section comprises the concepts, in terms of technology, sequencing and resources, to define a procedure to assemble each joint defined in the product section. Technology, sequencing and resources are collected in the work station concept, and work stations are grouped into the assembly line concept. The resources section comprises the concepts to define three main types of resources: jigs and tools, industrial means and human resources.

To implement the developed IDMU model, classes were mapped into elements of the commercial software (CATIA/Delmia V5). CATIA/Delmia V5 provides the Process-Product-Resource (PPR) structure to support the IDMU concept. The model is implemented by means of CATIA V5 macros within the application programming interface (API). The main result is an assistant tool, integrated within CATIA V5, which helps designers to generate FAL alternatives by defining scenarios and using knowledge rules, which are derived from technical staff's expertise. The application generates technological information integrated within an IDMU supported by the commercial PLM system. A very simple aircraft model was created and used to test the application. The results obtained in the executed case studies relate to requirements for: space, transport, resources, industrial means and cost; and allow validating the conceptual approach.

Defining an assembly process alternative, as proposed in the assistant application, requires use of the scenario information. It involves fixing an assembly sequence, establishing sub-assemblies associated to the sub-stages of the process, locating them into real industrial plants belonging to the set of available company's facilities, adding sub processes depending on the type of joint to be executed (e.g.: fuselage join-up) and assigning the resources to be used. Once the sequence is defined, sub-stages must be defined. Each one must contain a number of executions of joints and is related with a sub-assembly or set of components depending on the position of the involved joints within the sequence.

The next step is to assign sub processes to the work stations. A library, with a set of basic types of joints, was defined, where each basic type comprises the main sub processes to be carried out. Figure 20.10 shows the example of the Fuselage

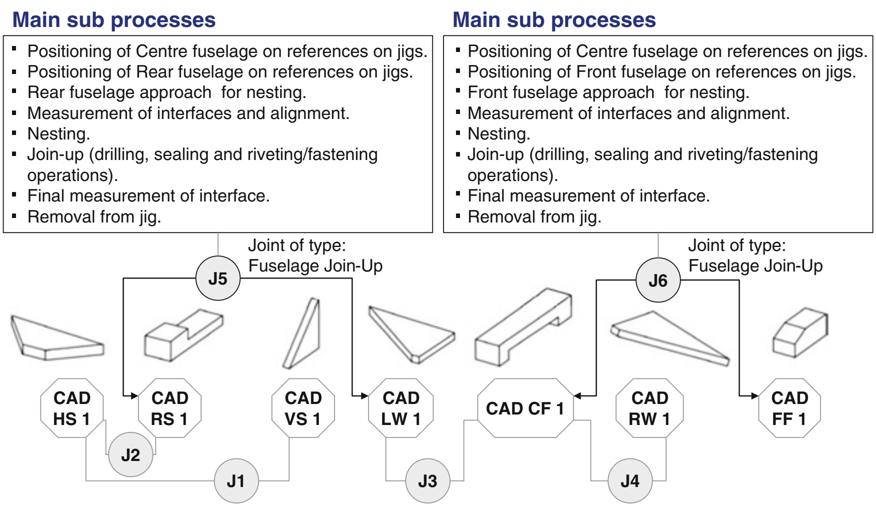


Fig. 20.10 Main sub processes for joint of type fuselage join-up [47]

Join-Up joint type. For each work station, depending on the type of joint to be executed in it, the design assistant, making use of the joint sub process library, automatically generates the main level of sub processes to be carried out and the corresponding nodes are created in the Process List structure provided by CATIA/Delmia V5.

The assignment of resources is the last step in the configuration of the process structure alternative. The designer has to select the resource type, input the value for each attribute and select the process node where the resource will be used. At this stage, a conceptual structure of a possible FAL solution is defined, and the designer is requested to select if more alternatives need to be evaluated.

Although the evaluation of alternatives is conducted at the industrialization conceptual phase, when products are still preliminarily defined, the evaluation of different scenarios allows creating estimates for different criteria that help in the decision making process. In addition to full implementation of the resource knowledge model, future work aims at implementing multi-criteria decision analysis and an automatic process planning capability in the form of an algorithm to create the 'as prepared' alternatives from the information defined in the 'as planned' structure, the joints to execute and the process information.

20.5.3 Multidisciplinary Design and Optimization

During the 90s, there was a trend to integrate structures and control disciplines into the early aircraft design process [14]. The complex aircraft system requires the coupling of the interacted disciplines, which then influences the performance of the whole system, where the optimal design can be facilitated by mathematical optimization. MDO has been widely implemented and adopted in aviation industry and frameworks with advanced optimization algorithms and KBE techniques have been built [15]. The MDO process is pushed to higher fidelity by coupling efficient analysis tools [14, 48, 49]. Technological fundament is explained in Chap. 4.

In recent years there has been an increased emphasis on integrating the structures and control disciplines into the design at an earlier stage [50, 51]. In structures, the increased use of advanced materials with their flexibility and reliability based design philosophies has been one driving force in MDO. One example is the deep coupling of powerful computational structural mechanics (CSM) and computational fluid dynamics (CFD) solvers. Another example is the use of composite materials for aeroelastic tailoring, as it couples structural detail (using skin fiber orientation angle) with the flexible wing aerodynamics and, ultimately, the aircraft performance.

20.5.3.1 Use Case: Fluid-Structure Interaction Simulation [52, 53]

For high-fidelity fluid-structure interaction simulations different tools are necessary to allow the highest possible accuracy. In this context the data transfer between the aerodynamic surface and the structural model, and the CFD mesh deformation are the key parameters for high performance due the high accuracy of modern CFD solvers. Therefore, the fidelity of these codes, which usually solve the Reynolds averaged Navier-Stokes (RANS) equations, is limited by the correct definition of the geometric boundaries. High fidelity models are not available in the early design phase of aircraft. Basic structural models, in which the wing is only represented by a beam, are often the starting point for fluid structure coupled simulations. In a later development stage more complex structural models are used which include a detailed representation of the lifting surfaces including control surfaces, but also of other aircraft components like the fuselage.

Here a coupling methodology is presented, which enables the combination of different structural representations in one coupling matrix. Different coupling methods facilitate the representation of aircraft components modeled with differing detail level. Detailed structural models, as well as beam structures and single-point representations can be treated in one method. Detailed finite element (FE) models are typically available for the wing, which allow to use radial basis function (RBF) interpolation, while the engines and flap track fairings are only modeled by single mass-points. Thus, only basic rigid-body splines can be used for the coupling of these parts. If the structural model is used in a high detail level, the size of the coupling matrix will get an issue in terms of performance and memory consumption. On account of this a comparison of an exported spline matrix and FSAdvanced-Splining, a fluid-structure-interaction (FSI) tool in the FlowSimulator software environment, is derived.

Afterwards an update to the mesh deformation module is presented, which enables to represent the exact deflections for every CFD surface grid node, which are delivered by the coupling matrix. Performance limitations do not allow to use all points as input for the basic radial-basis-function based mesh deformation method. Then the FSI-loop to compute the static elastic equilibrium is described and the application to an industrial model is presented. Finally, a strategy how to couple and deflect control surfaces is shown. Therefore, a possible gapless representation by means of different coupling domains and a chimera-mesh representation is shown. This section describes the bricks, which are combined to a fluid-structure interaction loop. Most of the tools are part of the FlowSimulator software environment (Fig. 20.11).

The coupling method allows to combine different interpolation methods for different model components. For the case of complex structural models with differently resolved components, this is a very important feature for fluid-structure coupling. Therefore, the structural and aerodynamic domain is spitted into several domains. These domains can be components, or further divided components to increase the numerical performance of certain interpolation methods. The technical integration of Nastran into the process is done via file exchange. Either binary or ASCII files are written and read to exchange forces and displacements. The data exchange of all FlowSimulator modules is done in memory.

The solving methods are combined to compute iteratively the static equilibrium state for certain aerodynamic target coefficients. The process loop is outlined in Fig. 20.11. The starting point for the solution sequence is the CFD-solver.

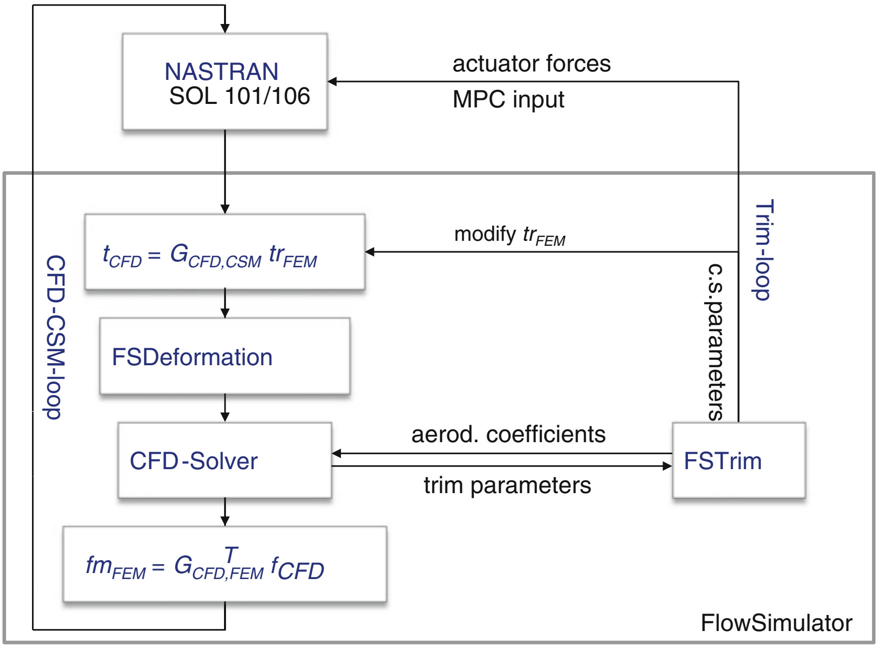


Fig. 20.11 Static fluid-structure interaction loop with additional trim loop

The diagram includes two loops, one for CFD-CSM-interaction and one for trimming. The trim loop begins after the CFD-CSM loop has reached a certain convergence level. Then the CFD-CSM-loop continues after the trim loop has fulfilled its convergence criterion. When both criteria are fulfilled, the elastically trimmed CFD-CSM solution is achieved.

Furthermore, it is shown that the trim module FSTrim computes parameter for the CFD solver like the different angles of attack, but also the control surface deflection angles (c.s. parameter). Depending on trim parameter, the trim loop continues with the CFD-solver, the displacement interpolation or the structural solver. This is necessary since the control surface deflection is handled on the structural node set. Either the structural deflection vector tr_{FEM} is modified by a rigid control surface deflection, or input is given to the structural model itself. For example actuator forces or multi-point-constraints (MPCs) can be used to change the position of the control surfaces. Actuator forces represent a force pair of equal magnitude but opposite direction, which is used to extend or shorten the length of actuator elements. An alternative way to model control surface deflections is provided by MPCs. Both allow to cover control surface deflections in the structural and aerodynamic domain. Additionally geometric consistency is assured. Another attribute of control surface deflections in the structural model is the advantage that the interpolation matrix can be used to take care of possible CFD-grid discontinuities.

As reference to the CFD-CSM-result a standard design tool result is used. The agreement of the two results is very good, only in twist a small deviation can be observed. The two introduced fluid-structure coupling methods did not show differing results. The coupling methodology allows the combination of different interpolation methods, each fitting to the boundary conditions of the used models. Since the spline matrix computes displacements for all surfaces nodes of the CFD-surface mesh, a correction algorithm for mesh deformation with RBFs is shown. As application example a complex aircraft example with a very detailed structural and aerodynamic model is presented. For the same test case the benefit of a "spline-on-the-fly" method is shown. It reduces dramatically the necessary amount of stored data for fluid-structure coupling. Finally, the flexibility of the coupling approach is underlined by giving some examples about the integration of a trimmed horizontal tail plane (HTP) and control surfaces into the coupling process.

20.5.4 Value Engineering

VE as a concept was developed at General Electric in the 40s on by Lawrence Miles as a method for considering the customer's willingness to pay for each element of added functionality in a product, where:

$$\text{Value} = \text{Function/Price} \tag{20.1}$$

VDD [54] is the process of optimising a product or service through a value function that best quantifies the value added of that product by following the steps of Definition, Analysis, Evaluation, and Improvement. Value operations methodology (VOM) [55] is an extension of the VDD approach with a focus on operational value that in turn requires optimal operations to be understood and utilised in the engineering evaluation process. VOM drives the design process with a more realistic operations based performance assessment that can pull better operational solutions into the market place. VDD and VOM rely on the use of a hedonic function, the typical form of the hedonic function of which relates the variation in cost to the variation in design characteristics, as presented in Eq. (20.2).

$$\ln(P_1) = \alpha_1 + \sum_{j=1}^{m} \beta_j x_{ij} + \varepsilon_i \tag{20.2}$$

where most importantly $j = 1 \ldots m$ is a set of value levers of the system analyzed, P is the price, and is a weighting factor associated to a defined value lever (or design characteristic) x. The value model is an evolution of Keeney's [56] representation of theorems for quantifying values using utility functions, as proposed by Fishburn, where Keeney defines Fishburn's function as the additive utility function as shown in Eq. (20.3):

$$u(x_1, \ldots, x_n) = \sum_{j=1}^{N} k_j u_i(x_i) \tag{20.3}$$

where u_i is an integral attribute utility function for attributes x_i, and k_i are the scaling constants that define a user's value system. Assuming that the additive utility function does not need to account for interdependencies relating to each consequence x, then there exists a corresponding magnitude of utility u that indicates the value [54]; as shown by Fishburn in 1965 [57]. The hedonic model establishes: (a) the Delta Price Principle: that it is reasonable to relate the price of one design instantiation to another and (b) the Additive Utility Principle: that the utility relating to a design instance can be simply accumulated according to the utility added by each feature or attribute. The VOM approach builds on both these principles.

Relative to (a) the Differential Principle, it is reasonable to assess the value of one design instantiation with another in terms of the value gradient relating to the value levers, resulting in a given delta value from the original state, whether positive or negative. This principle is further expressed in Eq. (20.4) relative to the value gradients:

$$\vec{\nabla} v = \vec{\nabla} f(x, y, z) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \tag{20.4}$$

where the value gradients are associated with a scalar function of the individual value functions or value levers (x, y, z). The Differential Principle suggests the use of both deltas and more fundamentally the gradients which give rise to the deltas. Equation 20.4 proposes that any value gradient or gradient vector field, $\vec{\nabla} v$ of the scalar function, $f(x, y, z)$ is indeed a function of the value gradients or partial derivatives; which are associated with the value model (as a scalar function) being a function of the individual value functions or value levers. Therefore, it can then be deduced that this can be expressed in terms of the standard vectors (I, J,K) associated with the individual value functions (x, y, z) and their partial derivatives, as shown in Eq. (20.5):

$$\left(\frac{\partial f}{\partial x} \hat{\mathbf{I}}, \frac{\partial f}{\partial x} \hat{\mathbf{J}}, \frac{\partial f}{\partial x} \hat{\mathbf{K}} \right) \tag{20.5}$$

Relative to the Additive Principle, it is reasonable to assess the value delta presented in Eq. (20.5) as an aggregation of all the individual levers' delta values. Consequently, this principle is further expressed in Eq. (20.6):

$$\Delta V(x_i, \ldots, x_n) = \sum_{i=1}^{N} \alpha_i \sum_{j=1}^{M} \omega_j \frac{(v(x_{ij}))_{\text{end}}}{(v(x_{ij}))_{\text{start}}} + \varepsilon_{ij} \tag{20.6}$$

where a change in value V is caused by a change in a set of associated value levers x_i, when moving from some initial state to some new state. Each value lever of the set $i = 1 \ldots N$ has an associated scaling factor a_i and error e_i and is in turn defined by a subset of lower level value parameters, x_{ji} *for* $j = 1 \ldots M$ and associated scaling factor ω_j, that describe the causal nature of each of each driver. The establishment of the lower level value parameter functions are carried out using the genetic-causal approach (GCA) presented by Curran et al. [18]. In short, this approach advocates modelling of value and cost by setting up families of products and establishing causal links between high-level cost drivers and its constituent elements.

20.5.4.1 Use Case: Applying VOM in Aircraft Design [55]

In the application of VOM to aircraft design, a model is proposed that captures the value of the aircraft design choices in terms of the operational impact and realisation through explicit value-adding criteria. The following value levers were utilised in a differential additive valuation manner as shown in Eq. (20.7). These value levers are subjective in nature and are to be selected by the user (as well as the weightings) but the authors included: Cost efficiency C (revenue/cost), Utilization U, Maintainability M, Environmental Quality E, and Passenger Satisfaction P. The methodology also proposes to use Safety S as a value lever as well as considering an error term. The differential principle is incorporated in the left-hand side of the equation while the additive principle is incorporated in the right-hand side of the equation.

$$\Delta V = \alpha_C\left(\frac{C_1}{C_0}\right) + \alpha_U\left(\frac{U_1}{U_0}\right) + \alpha_M\left(\frac{M_1}{M_0}\right) + \alpha_E\left(\frac{E_1}{E_0}\right) + \alpha_P\left(\frac{P_1}{P_0}\right) + \alpha_S\left(\frac{S_1}{S_0}\right) + \varepsilon \tag{20.7}$$

The influence of the value levers on each other is modeled with reference to Asavathiratham's influence modeling approach [58]. The value levers consist of the sum of specific system characteristics deltas multiplied by their associated weighing factors. The system-characteristic deltas are based on a reference aircraft's characteristics, correlating to those of the aircraft under consideration. For example, the *Cost* value lever is expanded as shown in Eq. (20.8).

$$\begin{aligned} C = {} & \omega_1 \cdot d[\text{DepreciationIOC}]\left(\frac{c_1}{c_0}\right) + \omega_2 \cdot d[\text{Tickets \& sales}] \\ & + \omega_3 \cdot d[\text{Admin \& other}] + \omega_4 \cdot d[\text{Staff}] + \omega_5 \cdot d[\text{Maintenance}] \\ & + \omega_6 \cdot d[\text{Fuel}] + \omega_7 \cdot d[\text{Crew}] + \omega_8 \cdot d[\text{Interest}] + \omega_9 \cdot d[\text{Insurance}] \\ & + \omega_{10} \cdot d[\text{DepreciationDOC}] + \omega_{11} \cdot d[\text{Airport}] \\ & + \omega_{12} \cdot d[\text{Navigation}] + \omega_{13} \cdot d[\text{PaxServices}] \end{aligned} \tag{20.8}$$

where C is the *Cost* value lever that represents the value score corresponding to the cost of the aircraft under consideration, w are the weight factors corresponding to the individual deltas, d[Depreciation IOC] is the delta of the cost depreciation for Indirect Operating Cost (IOC), d[Ticket/sales] represents the ticket/sales cost delta, d[Admin/other] defines the administration and other costs delta, d[Staff] is the staff cost delta, d[Maintenance] is the maintenance cost delta, d[Fuel] the fuel cost delta, d[Crew] Flight crew cost delta, d[Interest] is the interest cost delta, d[Insurance] defines the insurance cost delta, d[Depreciation DOC] defines the depreciation of the DOC delta, d[Airport] is the delta of the airport costs, d[Navigation] is the delta of the navigation costs and d[Pax Services] defines the passenger services cost delta. As mentioned, in implementation, the value model is based on a reference aircraft as a benchmark (subscript 0) relative to the performance data of the aircraft being designed (subscript 1), where the aim of the value model is to return the value of the aircraft under consideration relative to the benchmark aircraft. In essence, this is similar to the gradient based approach within optimization, where an improvement is sought rather than a specific level of value. However, the profound characteristic is that all value drivers are being taken into consideration and a balanced objective function is being used to find a more holistic global optimal.

Curran et al. [55] describe an application of the VOM model described above to a set of four aircraft types, being the Boeing 737–200, Boeing 737–800, Airbus A320 and Embraer ERJ-145. The top-level value levers were assigned weights as given in Table 20.1. For the individual lever weights, input values used for the specific value levers and value estimates, the reader is referred to Curran et al. [55].

The resulting estimates are highly dependent on the weights used, the accuracy of the used input values and assumptions such as linearity in performance characteristics, similarity in mission profiles, etc. Furthermore, this concerns a post hoc value analysis of aircraft performance. VOM may have further and more meaningful impact when used as a decision support tool in conceptual and preliminary aircraft design, when parametric estimates of aircraft operational performance may be used to investigate the value and consequently trade-off various competing design concepts. As such, VOM extends VDD by incorporating lifecycle considerations and is representative of the CE philosophy.

Table 20.1 VOM—airliner application—value model top level weights

Value in airliner design	Percentage
Cost	30 %
Sustainability	30 %
Market	10 %
Utilization	15 %
Maintainability	15 %

20.5.5 Life Cycle Costing [59]

Airlines globally are financially under cost pressure by rising fuel prices and introduction of CO_2 taxation schemes. In the past decade alone, the price of jet-fuel has quadrupled and the fuel component of DOC has increased from 14 to 30 % in 2013 [59]. With an increasing demand for jet-fuel and a reduction in global supply, the cost of fuel is expected to increase further.

Fuel consumption per passenger-km has already reduced significantly due to technological advances. The aviation is currently concentrating its initiatives on "drop-in" fuel solutions to achieve the necessary eco-economic transformation from petroleum derived Jet-A-fuel. The two major proposed solutions are biofuel and synthetic kerosene (Syn-Jet) made from natural gas/coal through the Fischer-Tropsch (FT) process. "Drop-in" fuels are currently being used experimentally in a blend with kerosene, but are still a long way from being commercially viable. Use of liquid natural gas (LNG), comprising upwards of 90 % methane, is already being used successfully in both automotive and maritime applications. It has also been explored as an aviation fuel, although LNG fuel applications have not extended to commercial fleets. Previous LNG feasibility studies raised questions over airport compatibility, safety and technology readiness levels (TRL).

To determine the impact of potential use of LNG, the LCC technique is used. This is the holistic analysis of the total cost of ownership (TCO) of an asset from its initial acquisition to its end of life disposal. It is typically used to determine the most economically rational option between competing alternatives that cannot be split based on technical appropriateness.

20.5.5.1 Use Case [59]: Life Cycle Costing of Alternative Jet Fuels

Transition to LCH_4 fuel will reduce airline DOC. Currently, fuel is 33 % of DOC and LCH_4 is less than 30 % of the cost of jet-fuel. This gap will widen as the cost of jet-fuel increases due to limited availability. Multi-national carbon emissions policies increase airline DOC. Environmentally, LCH_4 use will reduce CO_2 emissions by 20 % compared to jet-fuel, reducing carbon tax commitments. Consequently, the reduction in DOC will allow a reduction in fare prices, supports customer growth and increases income streams.

LCH_4 can be created from LNG or biogas generated from biological waste. This ensures a more sustainable supply of LCH_4 in the future and induces price stability. To assess airline DOC reduction from LCH_4 fuel use, an evaluation was conducted into the relative prices of competing fuels, the influencing factors governing these prices and the key impacts that may have on other aspects of airline DOC through stakeholder consultation and traditional research methods. Moreover, LNG is currently less than 30 % of the per energy cost of jet-fuel and promises to be available from untapped reserves of shale gas, harvested by the fracking technology.

To estimate LCC a modified approach on the TCO is used: every cost element of each technical alternative is assessed and summated for overall cost comparison; this approach assesses the particular cost elements that are deemed to have the greatest comparative impact on the overall LCC of an LCH_4 aircraft relative to a current baseline comparator aircraft. Additionally, contrary to traditional application, this report assesses the TCO from the perspective of the global commercial aviation industry in the event of a worldwide fleet introduction, as opposed to an individual aircraft acquisition by a particular transportation company. The three key cost elements that were seen to have a significant bearing on the relative TCO of an LCH_4 aircraft compared to a Jet-A kerosene baseline aircraft were identified as the cost of fuel for operation, the acquisition cost of the aircraft and the airport airline charges (which have been assumed as a worst case scenario where airlines shoulder the entire cost of infrastructure for a new fuel).

In order to provide an estimate of the comparative fuel costs for future years, the fuel prices for each year were estimated based on the percentage increase of the average yearly fuel price for the past 10 years (since 2003) for LNG (1.75 %) and kerosene (7.33 %). Whilst the extrapolation of the LNG price seems to align reasonably well with recent developments and the future outlook with the incorporation of shale gas reserves, it was highlighted that the continued rapid increase in the Jet-A Kerosene price projected may be more severe than the actual development. Therefore, to offset this, a more conservative projection, based on a projection of the future oil price provided by Airbus has also been included in all calculations.

Accounting for all cost components discussed (fuel, acquisition and new infrastructure), the total yearly cost savings by the introduction of LCH_4 aircraft compared to an equivalent number of baseline Jet-A kerosene aircraft for conservative fuel cost prognoses is depicted in Fig. 20.12. For the new airport infrastructure cost

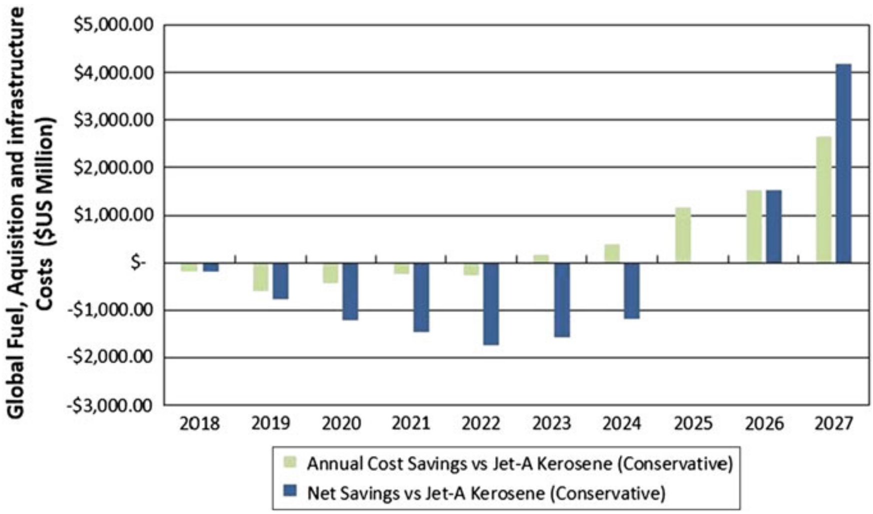


Fig. 20.12 Global fleet yearly fuel and acquisition costs [59]

component, there is no comparable cost incurred for Jet-A kerosene case as it is assumed that all required infrastructure is already in place for the new aircraft produced.

For the case that the Jet-A kerosene costs continue to rise at the same rate as the past 10 years, the aviation industry will run a relatively slight deficit before breaking even after 3 years and experience increasing savings. Alternatively, for the conservative prognosis of the Jet-A kerosene price, the breakeven point occurs 7 years after the initiation of the proposed global transition. With regards to the relative fuel, aircraft acquisition, and infrastructure costs, the aviation industry could make a net saving of US$4 billion to US$47 billion within 10 years if LCH_4 aircraft are introduced into the global fleet compared to the continued use of the Jet-A kerosene aircraft. This net saving represents 0.6–7.5 % of the total aviation industry's 2012 DOC from only a very small fraction of the global aircraft fleet. If the same rationale for LCH_4 variant was applied to other aircraft models on a larger scale, the savings would greatly multiply.

The design of LCH_4 aircraft alone is not a significant challenge as such aircrafts have been designed and operated in the past. The most significant upfront investment is in the infrastructure required for supply and storage of LCH_4 at airports. However, if the price of kerosene continues to rise as expected, conservative estimates show a breakeven in about 7 years after transition to LCH_4 is possible with a net saving of US$4 billion to US$47 billion within 10 years, if LCH_4 aircraft are introduced into the global fleet compared to the continued use of the Jet-A kerosene aircraft.

20.5.6 Systems Engineering

The main outlook of the SE approach has been defined at length in Chap. 9, as well as in sources such as the US Department of Defense (DoD) [18, 60] . Based on tasks integration and control as well as interfaces management, a requirements loop, which defines the iterative process between the requirements analysis and the functional analysis, forms the basis of the structure. The requirements loop identifies the relation between all the performance and other limiting requirements of the product. These requirements are used in the next iterative process which Curran et al. [18] term the 'design loop'. The goal of this loop is to move from a functional architecture towards a physical architecture. This is done by, trading off concepts, which are defined by configuration items, system elements and physical interfaces. SE is also adopted as a comprehensive, holistic approach to master the product complexity of complex products like aircrafts and foster the development of sustainable vehicles (see Chap. 27). It presupposes system thinking, in particular in design teams [61].

An important aspect of SE is the adoption of cost performance evaluation throughout the design process. Different costing methodologies are used throughout the design process such as the development of a cost breakdown structure (CBS).

The systems analysis and control has the task to monitor and manage all the aspects throughout the design process, that are needed for the technical analysis and the quantitative evaluation of alternatives (decisions made, requirements, risks, and others).

NextGen Air Transportation Systems (ATS) The Federal Aviation Administration (FAA) aims to transform the U.S. air transportation infrastructure from a ground-based navigation system to a net-centric satellite-based navigation system [62]. Due to the large number of involved stakeholders and high complexity of the project, the FAA has decided to introduce a SE approach. The project was initiated to anticipate on the increasing capacity of the navigation system and its side effects, which include an increasing number of delays and worsening of the aviation induced environmental impact.

In order to meet these goals, new technology needs to be integrated into existing systems at airports, aircraft and navigation system facilities. Besides the technological changes, processes and organizational structures need to be altered as well, to fulfill the requirements of the new system. Due to the transition from an isolated system towards a net-centric system, the verification and validation of NextGen requires a close collaboration of the involved systems.

The challenges above made the FAA chose a system of systems (SoS) approach. During the design process, multiple different development programs rely on each other to achieve the desired capabilities of NextGen. With 1820 FAA acquisition professionals working on 250 unique highly related programs, the FAA SE experience could be a pilot of high value for similar projects [61].

Network centric operations (NCO) occurs when systems are linked or networked by a common infrastructure, share information across geographic borders, and dynamically reallocate resources based on operational needs [63]. NCO recognizes that interdependence (sharing information among many) is vital to an organization's future. Information must be quickly distributed, its value understood and the desired effect created. NCO is an environment where seamless collaboration between networks, systems or elements within systems is possible. Understanding system-of-systems engineering (SOSE) is critical to a robust architecture development of NCO systems. There are five system-of-systems (SoS) characteristics but the dominating one is emergent behavior.

20.6 Conclusion and Outlook

This chapter has explored some the obvious coupling of CE within the integrated design approach within aerospace industry. Extended CE concepts such as CE have been discussed as well as some enabling concepts such as MDO and VE. Aircraft design, production and operation is a complex extended enterprise that demands life cycle integration and the compression of time without losing the fidelity of knowledge. MDO enables state-of-the-art integration of the CE process through tool development and integration into the business process. VE offers a radical view to the CE process in that the parallelization of tasks and life cycle requirements must be driven primarily with a view to what the ultimate value function or value goal is. The ultimate vision of design integration is to achieve concurrency in the integration of all relevant knowledge and to apply that to achieve the maximum with regards to the value that the product provides to the user. The VOM provides this component in particular in respect to what value the product or service adds. CE offers an encompassing approach to further developing these ideas and is long established in seeing them implemented by industry for value enhancement [64].

Future developments in the aircraft industry involve the introduction of new materials and/or material applications, new engine technologies, new control systems and evolution of the integration of the aircraft in the overall transport system. For the immediate future, the major OEMs (Boeing, Airbus) have chosen to design new iterations of their work-horse narrowbody aircraft families (e.g. the B737-MAX, A320neo), in essence representing an evolution of the conventional aircraft type.

The rise of new competitors (Embraer, Bombardier, COMAC) for the current market leaders Boeing and Airbus will enforce continuous consolidation in the entire supply chain like we have already experienced in the automotive industry during the past two decades [65]. Furthermore, common aircraft programs of two or more today's competitors can be expected in the near future which will induce additional complexity in the product creation process. Therefore, we will face additional challenges in the PLM of the extended aerospace enterprise (see Fig. 18.2). In comparison with the automotive industry (see Sect. 21.2), the product lifecycle of an aircraft is significantly longer than the lifecycle of any used software (PDM, CAD, etc.) and will, subsequently, set new requirements to the IT infrastructure in term of longevity, stability and scalability [66]. The process harmonization and standardization will get a significant impact. Long-term archiving and retrieval of product data, which is currently supported by the LOTAR International consortium, will, therefore, gain an increasing importance [67].

In the medium to long term, many adverse pressures on the aviation market will likely promote risk-adverse behavior in design, meaning that innovation on the overall aircraft configuration is likely to remain limited. However, on subsystem level, innovation will continue to be pushed as airliners are in a highly competitive environment where any saving is welcomed; on top of that, regulations (e.g. on emissions) are likely to strengthen the push for further innovation, in particular, if the current trend of continuously rising air traffic and even more aircraft in operations will be continued [3–6]. On the longer term, more esoteric designs such as blended wing bodies (BWB) may finally arrive in the civil aviation market.

References

1. Air Transport Action Group (2014) Facts and figures. Available at http://www.atag.org/facts-and-figures.html. Accessed 29 July 2014
2. FAA (2011) The economic impact of civil aviation on the U.S. economy, 2011-AJG-176, produced by ATO Communications
3. Deloitte (2013) 2013 global aerospace and defense industry outlook, expect defense to shrink while commercial aerospace sets new records
4. Deloitte (2014) 2014 Global aerospace and defense industry outlook, expect another record year for commercial aerospace and continued declines in defense
5. Airbus (2013) Global market forcast future journeys 2013–2032. Airbus S.A.S 31707 Blagnac Cedex, France
6. Boeing (2012) Current market outlook 2012–2031, Boeing Commercial Airplane Market Analysis, Seattle, WA98124-2207
7. VDI (2014) Boeing prognostiziert Nachfrage nach 36 800 neuen Flugzeugen. Available at: http://www.vdi-nachrichten.com/Technik-Gesellschaft/Boeing-prognostiziert-Nachfrage-36-800-neuen-Flugzeugen. Accessed 28 Aug 2014
8. NASA (2013) Green aviation: a better way to treat the planet. NASA Facts, Washington DC 20546
9. Eurocontrol Experimental Centre. (2006) CDM landside modelling—project phase 1: initial scenarios, from http://www.eurocontrol.int/eec/gallery/content/public/document/eec/report/2006/015_CDM_Landside_Modelling_Initial_Scenarios.pdf. Retrieved 13 July 2014
10. Aviation and Environment (2014) http://www.cleansky.eu/content/homepage/aviation-environment. Accessed 13 July 2014
11. International Civil Aviation Authority (ICAO) (2013) 2013 environmental report, http://cfapp.icao.int/Environmental-Report-2013/files/assets/common/downloads/ICAO_2013_Environmental_Report.pdf. Acessed 13 July 2014
12. Krein A, Williams G (2012) Flightpath 2050: europe's vision for aeronautics. In: Knörzer D, Szodruch J (eds) Innovation for sustainable aviation in a global environment. IOS Press, Amsterdam, doi:10.3233/978-1-61499-063-5-63
13. Fabrycky W, Blanchard B (1991) Life-cycle cost and economic analysis. Prentice-Hall, Inc., New Jersey, Chap 1, p 13
14. Piperni P, DeBlois A, Henderson R (2013) Development of a multilevel multidisciplinary-optimization capability for an industrial environment. AIAA J 51(10):2335–2352. Available at: http://arc.aiaa.org/doi/abs/10.2514/1.J052180. Accessed 22 Feb 2014
15. LaRocca G, van Tooren M (2009) Knowledge-based engineering approach to support aircraft multidisciplinary design and optimization. J Airc 46(6):1875–1885
16. Airbus (2014) Design offices and engineering centres. http://www.airbus.com/company/aircraft-manufacture/how-is-an-aircraft-built/design-offices-and-engineering-centres/. Accessed 13 July 2014
17. Aone (1991) AIAA technical committee on multidisciplinary design optimization (MDO). White paper on current state of the art. AIAA Technical committee
18. Curran R, Raghunathan S, Price M (2004) Review of aerospace engineering cost modelling: the genetic causal approach, progress in aerospace sciences, vol 40, no 8, Nov 2004, pp 487–534. http://dx.doi.org/10.1016/j.paerosci.2004.10.001
19. Grihon S, Krog L, Hertel K (2004) A380 Weight savings using numerical structural optimisation. In: 20th AAAF colloquium AAAF materials for aerospace applications, Paris
20. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manage 7 (3/4):324–347
21. Winner RJ, Pennell JP, Bertrand HE, Slusarczuk MM (1988) The role of concurrent engineering in weapons system acquisition, IDA R-338, Institute for Defense Analyses, USA
22. Bergstrom RP (1990) Assembly, promises and simultaneous engineering. Prod 102(2):50–56
23. Andreasen MM, Hein L (1987) Integrated product development. Springer, Berlin

24. Cleetus J (1992) Definition of concurrent engineering. CERC technical report CERC-TR-RN-92-003, Concurrent Engineering Research Center, West Virginia University, Morgantown, USA
25. Koufteros X, Vonderembse M, Doll W (2000) Concurrent engineering and its consequences. J Op Mgmt 19:97–115
26. O'Neal C (1993) Concurrent engineering with early supplier involvement: A cross-functional challenge. Int J Purch Mat Mgmt 29:3–9
27. Daft RL, Lengel RH (1986) Organizing information requirements, media richness and structural design. Mgmt Sci 32:554–571
28. Terwiesch C, Loch CH, de Meyer A (2002) Exchanging preliminary information in concurrent engineering: alternative coordination strategies. Org Sci 13(4):402–419
29. Chen Y-M, Liang M-W (2010) Design and implementation of a collaborative engineering information system for allied concurrent engineering. Int J Comp Int Manuf 13(1):11–30
30. Lu SC-Y, Elmaraghy W, Schuh G, Wilhelm R (2007) A scientific foundation of collaborative engineering. Ann CIRP 56(2):605–633. doi:10.1016/j.cirp.2007.10.010
31. VIVACE (2005) Collaborative methods and tools. VIVACE Forum, CERFACS, EADS, Warwick
32. Curran R, Butterfield J, Jin Y, Collins R, Burke R (2011) Value driven manufacture: digital lean manufacture, encyclopedia of aerospace engineering. Wiley, Hoboken
33. Bachmann A, Lakemeier J, Moerland E (2012) An integrated laboratory for collaborative design in the air transportation system. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering, 2013. Springer, London, pp 1009–1020
34. Dineva E, Bachmann A, Moerland E, Nagel B, Gollnick V (2014) New methodology to explore the role of visualisation in aircraft design tasks: an empirical study. Int. J Agile Syst Manage 7(3/4):220–241
35. Willaert SSA, de Graaf R, Minderhoud S (1998) Collaborative engineering: concurrent engineering in a wider context, a literature review. J Eng Techn Mgmt 15:87–109
36. Lenox M, King A, Ehrenfeld J (2000) An assessment of design-for-environment practices in leading US electronic firms. Interfaces 30(3):83–94
37. Todd S (1992) Collective action: theory and application. University of Michigan Press, USA
38. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Syst Manage 6(1):7–24
39. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manage 6(4):307–323
40. Boost Aerospace (2014) http://www.boostaerospace.com/Pages/home.aspx. Accessed 16 June 2014
41. Belkadi F, Troussier N, Eynard B, Bonjour E (2010) Collaboration based on product lifecycles interoperability for extended enterprise. Int J Interact Des Manuf 4:169–179
42. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int. J. Agile Systems and Management 7(2):101–115
43. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manage 7(2):116–131
44. Baumann O, Becker M, Doerfler I (2014) Coordination versus cooperation in interfirm collaboration: evidence from the airbus A350 program.In: DRUID society conference 2014, CBS, Copenhagen, June 16–18, http://druid8.sit.aau.dk/druid/acc_papers/mlo5b498y0gxt41puxguy069mmpv.pdf. Accessed 16 June 2014
45. Bondar S, Potjewijd L, Stjepandić J (2012) Globalized OEM and tier-1 processes at SKF. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 789–800
46. Bondar S, Ruppert C, Stjepandić J (2014) Ensuring data quality beyond change management in virtual enterprise. Int J Agile Syst Manage 7 (3/4):304–323

47. Mas F, Gómez A, Menéndez JL, Ríos J (2013) Proposal for the conceptual design of aeronautical final assembly lines based on the industrial digital mock-up concept. In: Bernard A, Rivest L, Dutta D (eds) Product lifecycle management for society. Proceedings of the 10th IFIP WG 5.1 international conference, PLM 2013, Nantes, France, 6–10 July 2013. Springer, Berlin, pp 10–19
48. Sobolewski M (2014) Unifying front-end and back-end federated services for integrated product development. In: Cha J et al. (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 3–16
49. Kolonay RM (2014) A physics-based distributed collaborative design process for military aerospace vehicle development and technology assessment. Int J Agile Syst Manage 7 (3/4): 242–260
50. Mainini L, Maggiore P (2012) Multidisciplinary integrated framework for the optimal design of a jet aircraft wing. Int J Aerosp Eng 2012(750642):9. doi:10.1155/2012/750642
51. Simpson TW, Martins JRRA (2012) Advancing the design of complex engineered systems through multidisciplinary design optimization. In: NSF workshop, 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, Honolulu, Hawaii, 23–26 Apr 2012
52. Stickan B, Bleeke H, Schulze S (2013) NASTRAN based static CFD-CSM coupling in flowsimulator. In: Kroll N et al (eds) Computational flight testing, notes on numerical 249 fluid mechanics and multidisciplinary design 123. Springer, Berlin, Heidelberg, pp 223–234. doi:10.1007/978-3-642-38877-4_17
53. Bleeke H, Stickan B (2013) Industrial ComFliTe applications. In: Kroll N et al (eds) Computational flight testing, notes on numerical 249 fluid mechanics and multidisciplinary design 123. Springer, Berlin, Heidelberg, pp 249–256. doi:10.1007/978-3-642-38877-4_17
54. Collopy P, Hollingsworth P (2009) Value-driven design. AIAA Paper 2009–7099. American Institute of Aeronautics and Astronautics, Reston, VA
55. Curran R, Beelearts W, Abu-Kias T, Repco MJF, Sprengers YLJ, van der Zwet PNCS (2011) Value operations methodology (VOM) applied to medium-range passenger airliner design. J Aerosp Oper 1(1):3–27
56. Keeney RL (1988) Building models of values. Eur J Oper Res 37(2):149–157
57. Fishburn PC (1966) Stationary value mechanisms and expected utility theory. J Math Psychol 3(2):434–457. ISSN 0022-2496 http://dx.doi.org/10.1016/0022-2496(66)90023-X
58. Asavathiratham C (2000) The INflUENCE MODEL: a tractable representation for the dynamics of networked markov chains. PhD Thesis, Massachusetts Institute of Technology, pp 2
59. Conroy T, Bil C (2014) Life cycle costing for alternative fuels. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc international conference on concurrent engineering, IOS Press, Amsterdam, pp 92–101
60. Department of Defense (1999) Systems engineering fundamentals, Defense Acquisition University Press, Fort Belvoir
61. Lamb, CMT (2009) Collaborative systems thinking. An exploration of the mechanisms enabling systems thinking. PhD thesis, Massachusetts Institute of Technology, Cambridge
62. Federal Aviation Administration Next Generation Air Transportation System, (May 30 2014). Retrieved 13 July 2014, from http://www.sebokwiki.org/wiki/Federal_Aviation_Administration_Next_Generation_Air_Transportation
63. Hsu JC (2014) In a network-centric World. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc international conference on concurrent engineering, IOS Press, Amsterdam, pp 17–25
64. Zorgdrager M, Verhagen WJC, Curran R (2014) An evaluation of forecasting methods for aircraft non-routine maintenance material demand. Int J Agile Syst Manage 7(3/4):383–402
65. Mas F, Menéndez JL, Oliva M, Ríos J (2014) Collaborative engineering: an airbus case study. Procedia Eng 63(2013):336–345

66. Katzenbach A (2010) Engineering IT heute – Wege in die Zukunft. Arbeitskreis Softwaretechnologie der Region Bodensee, Konstanz, 12.11.2010, http://akswt.uni-konstanz.de/docs/2010-04_Konstanz.pdf, Accessed 15 March 2014
67. NN (2014) Long term archiving and retrieval—LOTAR. http://www.lotar-international.org/lotar-standard/overview-on-parts.html. Accessed 15 Aug 2014

Chapter 21
Automotive

Alfred Katzenbach

Abstract The automotive industry is one of the most advanced industries using information technologies for product development. The product variety and complexity have grown dramatically over the last decades. These enhancements could only be achieved by using the full range of technologies and methods described in part two. Within automotive engineering companies are continuously looking for new ways to achieve economic growth. Trends show that this is often done by expansion of existing markets as well as entering new markets, providing niche products and increasing productivity. This effects significantly the continuous development of processes and IT solutions. Legacy Systems have to be integrated with modern solutions. Service oriented architectures (SOA) and semantic nets will lead to a new system landscape. This change is not only a technical one but also an organizational paradigm shift which has to be handled carefully. To establish an international, multi-company concurrent engineering process, a common understanding of processes and business objects is required. The most efficient way to do this is standardization. The "Code of PLM Openness" (CPO) helps to find a common definition which lead to a better understanding of system integration and usage of standards. Two Standards play a significant role: ISO 10303 (STEP) with its new application protocol 242 which combines the known protocols for automotive and aerospace including model based system engineering and ISO 14306 (JT) for DMU and geometrical collaboration. The continuous enhancements of CAD systems lead to a knowledge-based engineering (KBE) approach by handling parametrics and associativity.

Keywords Product concepts · Automotive development system · Simultaneous engineering · System landscape of engineering IT · Semantic nets · Service orientated architecture · Code of PLM openness · ISO 10303-242 · ISO 14306 · Knowledge based design

A. Katzenbach (✉)
Katzenbach Executive Consulting, Gaienhofen, Germany
e-mail: alfred@katzenbach-web.com

A. Katzenbach
University Stuttgart, Stuttgart, Germany

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_21

21.1 Introduction

Satisfying mobility, one of the most expressed human needs, the automotive industry has been one of the main drivers in product development as well as in industrial production during the last one hundred years. With a yearly production of more than 70 million passenger cars and 12 million trucks, it is a leading economic factor in most of the industrial nations. Accordingly, the automotive market is highly competitive everywhere. The competition is fought out between about 10 big manufacturers in the passenger car and truck segment. Although the product car has long since become part of the "old economy", product innovations remain essential for the success of the automobile industry [1]. Starting from optionless mass production the tendency has developed more and more to a huge variety of different vehicle variants combined with configurations. To achieve and to be able to handle this complexity and variety, significant changes in product concepts, supplier partnerships and internationalization have to be accomplished.

Driving forces for these changes were accompanied with crucial shifts in organization, processes, and methods of automotive development. At first, this chapter addresses the current developments of automotive industry (Sect. 21.2), followed by a description of automotive development system (Sect. 21.3). The basic infrastructure of IT systems for automotive development is highlighted in Sect. 21.4. Focus on concurrent engineering is set in Sect. 21.5. The crucial role of international standards and supplier integration is drawn in Sects. 21.6 and 21.7. Application of knowledge-based engineering (KBE) is described in Sect. 21.8, followed by conclusions and outlook in Sect. 21.9.

21.2 Development of Automotive Industry

Addressing new customers in global context, the automotive industry has adapted all relevant business processes to meet new challenges [2]. The current winners in the global competition found their market position on significant improvements in product concept, supplier integration, and globalization.

21.2.1 Product Concept

In the twentieth century, the product portfolio was mostly structured in independent carlines with individual parts. Parts with similar purposes in different carlines were different and developed independently. The variety of selectable options increased over time—but slowly.

Looking to the best known vehicle of the 50s and 60s, the VW Beatle, it was produced in only one variant, with one engine size over 25 years. In this time, only minor changes to the exterior, interior, and powertrain were introduced.

The tendency towards more individuality leads to concepts with common body architectures (platforms) and reusable modules (Fig. 21.1).

The successor VW Golf was launched in 1974 and currently the 7th version is in production. Over this time, the concept has been developed from a single vehicle over a platform as base for a variety of cars to the current “MQB”, a modular kit of car bodies which is the foundation of large number of different models of the brands Volkswagen, Audi, Seat, and Skoda (Fig. 21.2) [3]. This concept is still in the

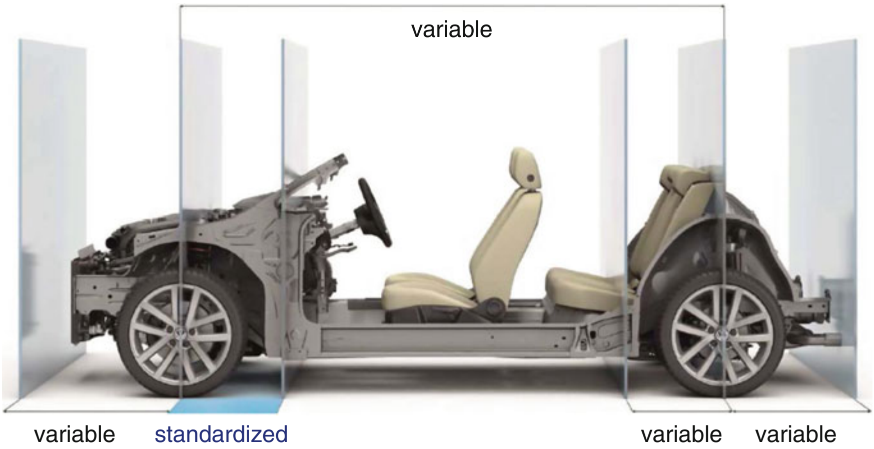


Fig. 21.1 Basic architecture of modular kit “MQB” at Volkswagen [3]

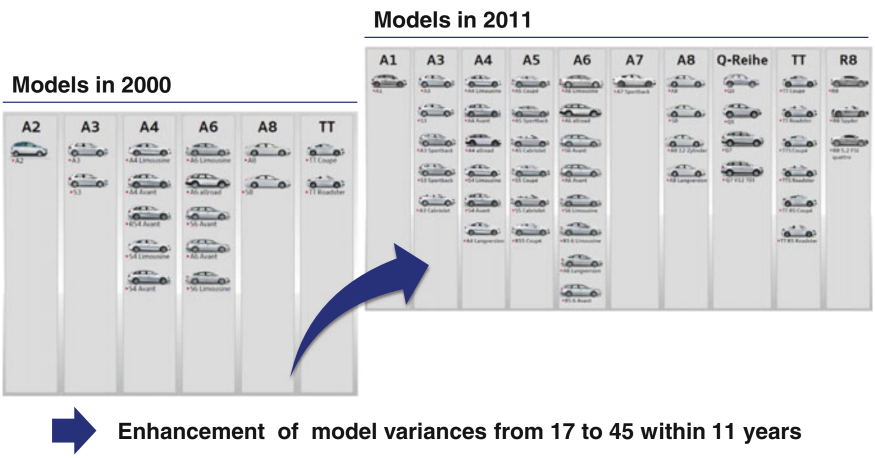


Fig. 21.2 Development of vehicle portfolio of Audi within 11 years [4]

implementation phase and defines new requirements to the supporting IT-systems because of its variety of car bodies, configurable options, different engines, and country specific characteristics.

Similar to the variation concepts on the body side, the modularization took place on the powertrain. Due to the high development effort and time required for developing a new engine or gearbox, it is the intention to adapt the same engine into a maximal number of different car models. In addition other important drivers are the intensive effort to reduce fuel consumption and the deployment of the cars in different countries. Beside the availability of fuel quality the biggest drivers are the legal regulations. Therefore, a concept of a maximum of reusable basic parts (core engine) in combination with different car bodies and regulation related mounting parts was established. Additionally, the interface between the engine and the gearbox has been standardized allowing a flexible combination of engines and gearboxes and potentially hybrid modules.

Figure 21.2 shows impressively the opportunities which can be gained with such an approach. This tremendous extension of ranges and models was achievable without a proportional growth of manpower and budgets.

At the same time, “time to market” is more and more important. This leads to the tendency to reduce the development time from product definition to start of production. In the 70s and 80s development cycles could go up to 10 years, whereas the current trend is to achieve cycle times of 30 months or less. To deliver a product at the right point in time contributes directly to the revenue. If a car has a delay of 6 months, the lifecycle revenue declines by 30 % [5]. Aligned to the reduction of development time, the production lifecycle was approximately reduced by 30–50 % during the last 30 years.

21.2.2 Supplier Integration

In the value chain of product development, the contribution of supplier has also changed. The suppliers have developed themselves from engineering partner to system developer or system provider, who develop, produce, and deliver dedicated systems to automotive OEMs [6]. The largest suppliers often state, that they produce almost the entirely car as parts and the OEM primarily conduct the integration and assembly. These enhanced capabilities will cause a higher impact in the future when we talk about mechatronic systems with intensive dependencies between formerly independent components.

According to a study conducted by the German Association of Car Manufacturers (VDA) at the end of 2012 [8], the proportion of global value added in R&D contributed by automobile manufacturers will drop from the current value of 60 to 47 % by 2025. During the same period, the contribution made by suppliers will increase from 32 to 36 %. The proportion due to engineering services will increase from 8 to 17 %. At the level of production, the proportion of value added attributable to suppliers will increase from 65 to 71 %. This means that although Europe will continue to lead the way at the level of R&D, China will become the largest center of production. According to the Bank of America Merrill Lynch [7], the top five revenue drivers for automotive industry suppliers alongside production are global expansion, the acquisition of new customers, enhancements to component functionality (e.g. entire cockpit instead of single instrument), consolidation (acquisitions) and outsourcing by automobile manufacturers. An example on product level is impressively illustrated by Fig. 21.3. It implies worldwide distributed engineering and supply chain with high level of collaborative engineering (see Chap. 7), supported by powerful IT technology. It facilitates fulfillment of specific customer needs in each local market combined with cost-optimal operations and high flexibility. Pre-requisites are global PLM concept (see Chap. 16) which preserves ubiquitous information management, and appropriate intellectual property protection (see Chap. 18).

Of course, the automotive industry is just one example for this trend. The aviation industry with its extended enterprise initiatives is facing a similar challenge.

21.2.3 Globalization

The market penetration of vehicles in the established developed countries has reached a level that generates no additional opportunities for significant further growth. In fact, it seems to be the other way round. In the last decades, the emotional

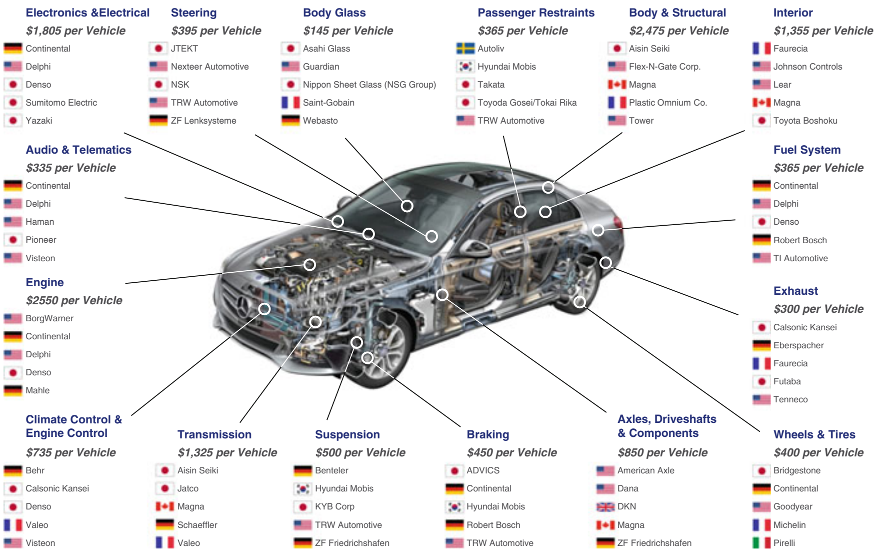


Fig. 21.3 International contribution of suppliers in the automotive value chain (according to [7])

value of ownership of a car has decreased in those countries and is further going to change into a direction with a flexible usage of transportation services instead of owning a car as a status symbol.

This tendency is totally different in the emerging markets. The penetration of ownership of vehicles is much lower and holding an own car is desirable. However, these emerging markets such as China, Brazil, India, or Russia have their own rules and the potential customers are interested in options and characteristics of the cars which are often not well understood in the established car developing countries. To understand and meet this demand, it is crucial for each OEM to establish own development facilities in these countries to understand the market properly and conduct specific developments and adaptions which meet customer demands. The local development sites give the automotive OEM also the chance of hiring high talented people from the different countries who have a much better understanding of the local requirements.

On the other hand, there is also a significant demand for localization of production. The emerging markets need to participate from the value added inside the countries. It is also in the interest of the international OEMs to avoid currency risks by producing in the countries where they sell the products. Apart from that it is even requested from the government of these countries to increase the local content of value added production to prevent toll barriers and other obstacles.

These tendencies in globalization of product development as well as production in the countries, where the cars are distributed require IT solutions for global engineering (see Chap. 7) collaborations which allow them to work seamlessly, respecting country specific differences and ensuring the individual interest of intellectual property protection (see Chap. 18).

21.3 Automotive Development System

Developing a new car model takes a timeframe of 3 up to 5 years involving some thousands of individuals in the process. To make such an effort manageable and ensure that the each output becomes traceable, it is necessary to install a clear process to describe the different roles, tasks and deliverables to the dedicated steps of the process.

This concept of a generic development system harmonizes the development process over all development facilities and all product lines, and ensures, that at the end the targets regarding quality, time, and budget can be achieved. This includes not only the classical development part of the activities but also the relevant contribution of production, purchasing, marketing and sales, services, quality management, human resources, and financials.

The concept of an automotive development system is oriented to the internationally known principles of project management, often represented by the international "Project Management Institute" (PMI) [9]. Main properties of these systems are:

- Standard project organization with clearly defined structure, roles, and responsibilities
- Master process plan (tasks with clear input and output descriptions, dedicated roles)
- Defined milestones over time to harmonize the parallel process chains.
- Key performance indicators (KPI) to get a traceable project status over time

Beside this process definition, the development system also defines a standard project organization which ensures that all parties necessary to contribute to the project success are involved in their specific role.

In addition, it is a process definition which also safeguards the efficient contribution of external partners to the development process.

21.3.1 Standard Project Organization

In the standard project organization all disciplines such as car body, exterior, equipment, interieur, electric/eletronics, chassis. engine, powertrain, vehicle integration, cross divisional functions are represented. For each discipline a project lead is appointed and has to take care about content, quality, time, and cost. Typical roles are (amongst others):

- Project owner with prime responsibility for the project
- Driver, who is responsible for recommendations to decision making
- Project manager, who is responsible for project execution
- Individuals, who actively work in the project team
- Additional experts for project issues.

21.3.2 Master Process Plan and Quality Gates

Based on a multi step approach a master process plan defines tasks for all involved disciplines in the development process. The individual targets that have to be fulfilled are defined. Each step will be finalized by a quality gate, where all partners have to report the current status which is requested at this point in time. To simplify this reporting a status report based on traffic lights is often used.

Three status colors can be reported referencing to the value of the defined performance indicators:

- Red: the actual value lies outside the agreed target. No validated actionplan is available
- Yellow: the actual value lies outside the agreed target. A validated action plan is available
- Green: The actual value satisfies the agreed target

The assessment of project targets includes the explanation of performance indicators, the needed counter measures in case of “yellow” and “red”, and degree of fulfillment as well. At the Quality Gate, all disciplines report to an Executive Committee, which can make the required action plans and decisions to fulfill the overall project target.

21.3.3 Milestones in an Automotive Development Process

Each automotive manufacturer has their specific development system and the details of this system depend on past experiences, product variety, and company culture. Respecting the tendency described (Sect. 21.2.1), it is requested to distinguish between the development of a new architecture of a new model family or an extension of such an architecture. The number of defined milestones also differs from OEM to OEM (Fig. 21.4).

The description shows a 10 step approach which gives a general overview of developing a new car architecture with its first extension (Fig. 21.4). For additional extensions it can be reduced [10].

To compare development time between different Automotive OEM’s, the time spanning between step 6 and step 1 is taken into account.

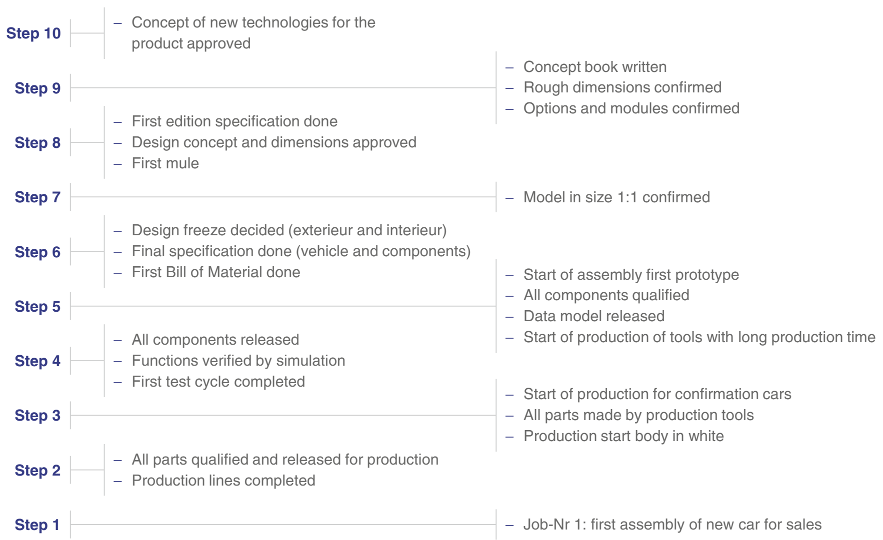


Fig. 21.4 Milestones in an automotive development process

21.3.4 *Generic Development Landscape for Mechatronic Components*

In addition to the Vehicle Development System of Sect. 21.2 for each domain responsible for a component, system or the entire vehicle, a generic multi-disciplinary development plan is in use (Fig. 21.5).

In the past the disciplines of mechanics, electrics/electronics, and software acted more or less independently. Today this is not possible anymore due to the huge number of dependencies between the three domains [11]. Modern automotive product development has to follow a systems engineering approach with a clear structure of requirements, functions, logics and physics (see Chap. 9) often described as "V model". Overall, this approach facilitates a broad interoperability of creative (design) and analytical (test) activities during the product creation process. Furthermore, the V model allows subdivision in multiple layer (mechatronic, mechanic, resource, application, etc.). It encompasses the specification of the complete product (vehicle, system, device) broken down into parts specification, design, and evaluation of parts on the left leg of V. Upwards on the right leg of V lie the validation and aggregation of components, systems and entire product. In the same manner this approach is used in collaboration context between OEM and suppliers. Systems engineering offers rich methods supporting mastering of complex systems: Manifold architectures, methods for simulating and testing vehicle properties during the development phase, and methods for developing processes belong to the toolkit of systems engineers [12]. Functional safety gains its growing importance as distributed mechatronic systems become more and more responsible for safety-critical properties.

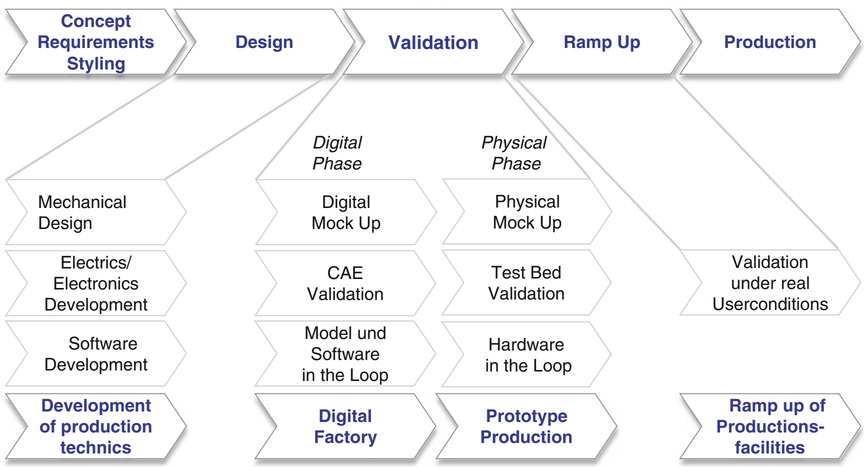


Fig. 21.5 Generic multidisciplinary development plan

21.4 Basic Infrastructure of IT—Systems in Automotive Development

Figure 21.6 shows a generic landscape for automotive development [13]. This landscape is represented by different systems in the development process. Depending on the automotive OEM, its history, locations and way of working, OEM's have either a single system approach for each domain or a multi system approach. It is also possible to run different domains combined in one system. In this landscape we can find home grown proprietary systems, systems from small highly specialized IT vendors as well as big systems from global IT vendors used in multiple companies. Often the big systems are highly customized and at the end more or less similar to proprietary systems. The way of using the systems in a company environment is a major part of the specific knowledge inside the company. This is the reason why it is so difficult to change or consolidate the system environment. Processes supporting software solutions at an automotive OEM have a long-living position and takes immense effort and time to change.

The quality of process integration depends on the ability of the systems involved to interact with each other seamlessly. Here technologies such as service oriented architectures (SOA), WEB 2.0 capabilities, and international standards like ISO 10303 (STEP) and 14306 (JT) support this integration. If new solutions have to be added to the system environment, the ability of integration plays a major role. This is the reason why the automotive industry started the initiative to define a "code of PLM openness" (CPO) to evaluate and classify potential solutions concerning the ability to integrate (see Sect. 21.5).

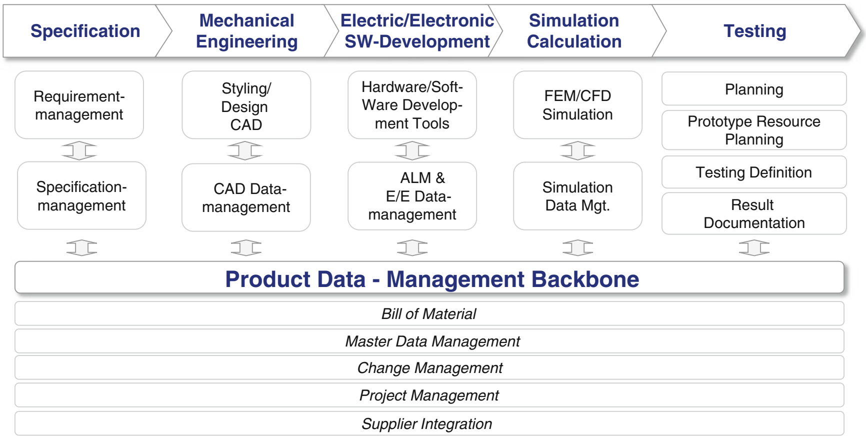


Fig. 21.6 Generic system landscape for automotive development [13]

It's backbone is the product data management system [14], containing at least

- Document management
- Configuration management
- Version management
- Release management
- Rules- and rights management

Other functions such as project management, workflow management, long term archiving, CAD data management can be included. The PDM backbone also has to control the different team data management solutions and is the source for all following downstream processes like Bill of Material, digital factory etc.

The ability of a larger automotive company to integrate the different modules of this landscape generates a competitive advantage in the product development process. It is not only a question of technical integration. The main work to be done is the clear and stringent tailoring of the data objects and user functions to the different modules. Today's landscape in automotive companies represents a mix of home grown legacy systems, commercial systems which are more or less intensively customized and new modern applications based on modern IT frameworks. Especially the homegrown systems have a long lasting history and a difficult to control number of interfaces. For example a Bill of Material System can have about 500 interfaces to other systems running in the different business processes inside a company. A new implementation for replacing such an old system is normally not a possible approach. Besides the tremendous effort and high cost which results in nearly no chance of finding a positive business case the risk of negative impact during the change phase is very high. The Bill of Material system manages the nervous system of a company and there are good reasons to call the PDM System the backbone system of product development. Analogous to human beings, surgeries involving the nervous system or the backbone are highly risky. Experiences of different projects at some international OEM's show that such a change takes more than 10 years from project start to finalizing the rollout and some hundred million Euros project costs. Quite often, such projects fail and are stopped without any rollout.

The difficulty to manage enhancements raises with the number of systems involved. The continuous development of processes requires continuous enhancement of the IT systems. By implementing these requirements the dependencies of the systems and the effort for comprehensive testing and roll out increase disproportionately.

To tame these challenges and reduce the risk and effort involved, new concepts such as SOA are required. From the theoretic approach, SOA aims to create a cost optimized and easy to maintain IT environment. On the way of implementing SOA it is more and more obvious that the SOA approach is the foundation for a step by step, cost optimized and risk minimizing way to renew an established IT infrastructure. The way of a continuous transition from an old stable, but inflexible and cost intensive environment, into a new flexible and future oriented landscape is more than a project or a program, it is a continuous journey that takes more than one decade [15].

To ensure the success of such a journey a future oriented, flexible, and expandable concept has to be developed and maintained continuously. Beside the technical concept it is very important to get the required buy in from the IT specialists, who have to realize this journey, and a clear long term oriented governance structure, which takes care of the process, the developed solutions and the continuous change of involved people over such a long period of time [16].

21.4.1 Principle of a SOA Based Integrated System Architecture

The guiding principle to realize of SOA-based approach is a common, system independent engineering client (CEC) allowing the definition of a "Role based Workplace" [17]. Each participant or user group in the complex development process has different tasks to do and has to use a variety of systems. In general each user is only using a limited number of systems actively for authoring or execution, but most of the systems only passively for viewing and as source of information. Today's systems are normally not structured in a way that appreciates the different user scenarios. With the concept of a role based client it is possible to configure the system environment in a way that it is oriented to the business demands of different user groups and can support the different processes efficiently. To find the right balance of user orientation and customization effort it is required to limit the number of roles to a range between 10 and 20.

The common engineering client (CEC) gets its data and business logic from an engineering service bus "ESI" which builds the middle layer between CEC and the under laying legacy systems (Fig. 21.7).

To design the path from the system orientation of the past to the new service orientation, a clear tailoring and alignment of the data objects and business logic to the backend service provider has to be organized by a service oriented domain model which covers all business processes and the relations between the single domains. This domain model is in general based on the landscape of Fig. 21.6. This domain model also gives the orientation for a step by step migration of the backend systems to become SOA capable.

21.4.2 SOA Based IT Organization

The step by step migration of an established system environment into a new service orientated world is not only a technological and procedural challenge. It requests also a clear focus on the IT organization and the people working in such an environment. In the past, IT teams had a dedicated responsibility for single systems and got their work identification from the pride of being accountable for the

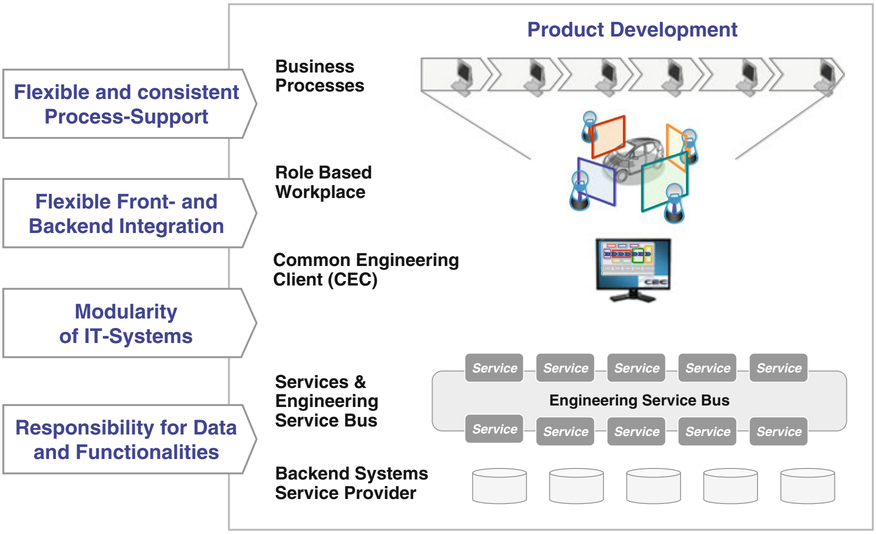


Fig. 21.7 SOA based integrated system architecture [17]

capability and process performance of the specific system. Over time the teams could develop an identity for themselves creating a spirit which potentially generates incredible performance and willingness to succeed. This is especially true in critical situations such as release-changes or major incidents. This kind of "in group behavior" is a characteristic of high performance teams which generates familiarity, motivation and high empathy for team internal cooperation. Such a incommitted team can achieve unimaginable results. In such teams the alignment to the team success has for the members a higher value than their individual success.

However, the risk of such high performance teams is the tendency to be capsulated from others teams and leading to difficulties to collaborate with others. The migration in a service oriented world destroys this long term grown structures by the request of giving up the independency. The degree of freedom for system specific decisions is reduced and the impression is a kind of loss of self-control. The members of the groups feel a reduced appreciation of their individual expert status. Additionally, they are required to collaborate with members of different teams which were seen as enemies before. This leads to an intensifying of competition fights and demotivation in line with a significant loss of productivity [18].

To overcome this situation and reestablish a high performance of the complete IT team, a structured organizational change process is mandatory in which the frozen structures have to be melted down in order to generate a mindset in which changes are possible. In this process, a plant destabilization with an intensive individual suffering has to be run through, before it is possible to create new "meta in groups" which are capable to collaborate constructively and productively in the new organizational environment (freeze-unfreeze-freeze-paradigm).

This mental change program must be done in parallel to the regular work. In this phase strong leadership is required. This has to be aligned with clear rules. These rules should be provided and controlled by a governance structure which takes care of all development and maintenance activities [16].

At the top of this governance structure is a joint executive board of all senior executives, meeting at least once a month on this topic. The executives have to define the guiding principles and control the continuous implementation. In case of conflicts the group has to moderate and find a consensus or make a clear decision.

Under this executive board, different boards on manager levels have to take care of the different steps. Each board has to be joined by a member of each former system oriented team to ensure, that all aspects of the different former organization are considered.

The installed boards are

- Requirements Board: All requirements have to be defined as process requirements. Requirements for dedicated systems will be rejected. The board receives the requirements, does formal checks and quality validation. If a requirement is accepted, it gets a req-number. Further on it will be controlled over the lifecycle until it is closed.
- Process Board: Takes over the quality controlled requirements and works out the consequences for the different development processes. Side effects as well as contradictions in other process areas are evaluated. The changes in the process will be defined as well as the effects of the new process on time, cost and quality. At the end, the requirement will be allocated to one or different domains of the engineering domain model.
- Architecture Board: Takes over the domain allocation of the requirement and tailors it into services. It defines the way of implementation in one or different systems and decides the way of frontend (CEC) or backend (ESI) integration. In addition a cost and time calculation for implementation is done.
- Client Board: Analyses the consequences and demands of the process change on the client side and defines requirements for the client enhancements including cost estimation.
- Release Management Board: Consolidates the results of Process-Architecture- and Client board. The business case for the requirement will be calculated and the implementation order released or rejected. The requirement will be aligned with one of the planned releases and further tracked over the implementation cycle and at the end released in the planned IT Release.

21.4.3 Quick Wins Based on SOA

Beside the general strategic conception it is very helpful for the long-term success to generate some quick wins. These give the user community the chance to understand the foundation of the complex technical approach and that it creates some concrete benefits. These increase the support of the program and can play an important psychological role

One example to leverage benefits to the user might be a visualization service for 3D data and drawings, which can be used in multiple user-scenarios and so expand the potential usage of data stored in the PDM system significantly. Without such a service, 3D data and drawings can only be used by people who are familiar with CAD or PDM Systems. A visualization service provides an easy to use access to such data. The goal of the service is to provide a read only access to parts and assemblies with a simple viewer running in a normal office environment (Fig. 21.8). So people without specific CAD/PMD know how can view and analyze this data. It can be integrated into many business processes. If a part or assembly number is displayed in an office document like Word, Excel, PowerPoint or even in SAP or other applications the user can do a right click on the part number and directly select the related 3D data or the drawing in an additional window to operate some limited functions. Obviously all rules of access rights will be taken into account by using appropriate services ensuring, that only those people who are permitted get access.

21.4.4 Roadmap to a Role-Based Workplace

A CEC to establish a Role-based Workplace introduced in Fig. 21.6 can also implemented in a step by step approach. This is not only relevant for the IT-implementation but also for the change management of the users. The requirements profile for CEC has two major, in the first view contradictory demands. On the one hand it should have a homogeneous system independent and integrated user

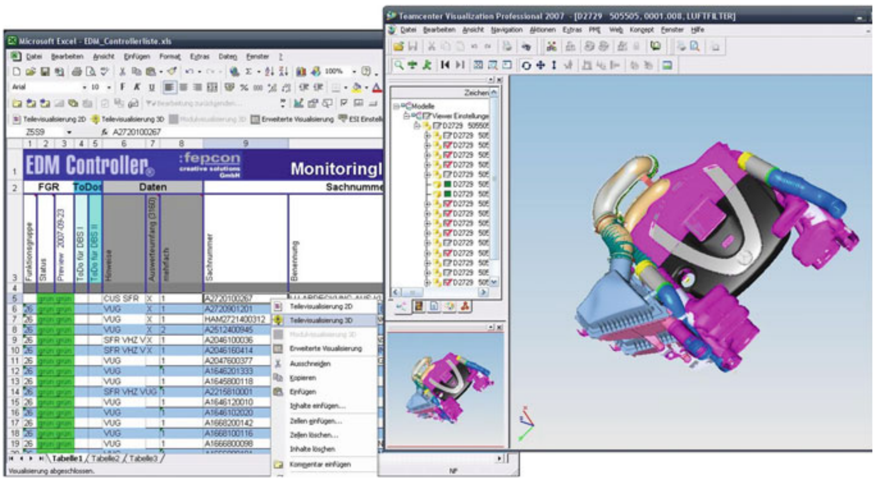


Fig. 21.8 Visualization service in office environment [17]

behavior; on the other hand it should respect the specifics and methodical autonomy of the different applications to be integrated.

For the user, the change to this new concept is an organizational change comparable with the change in the IT team mentioned before. So the involvement of users during the implementation phase and the training in the role out phase have a very important contribution to the success of the initiative.

In the first step, the foundation of the framework has to be developed, including style guide, user model, implementation rules, security regulations, printing, etc. In this phase the principles of implementation with first technical integration and access to basic services such as a central identity management enabling a single sign on or rights management have to be developed and approved by a number of prototypes, which have to be evaluated by the user community (Fig. 21.9).

To give the users the chance to gain their first positive experiences with this new client concept, it is helpful to implement dedicated system perspectives first. So the users can then be trained with the new interface and learn the new client behavior.

In the phase of implementation of the system-perspectives the following steps of client integration and backend integration should be considered as much as possible. For client integration the framework has to provide some generic services. The effort for client integration is much lower compared to backend integration. On the other side backend integration is requested because of the size of the client environment. The frontend integration has a direct relationship to the growth, performance and flexibility of the client environment. Hence, it is a natural step to do frontend integration first, gain experiences in user situation and then bring the integration services into the backend in a later release.

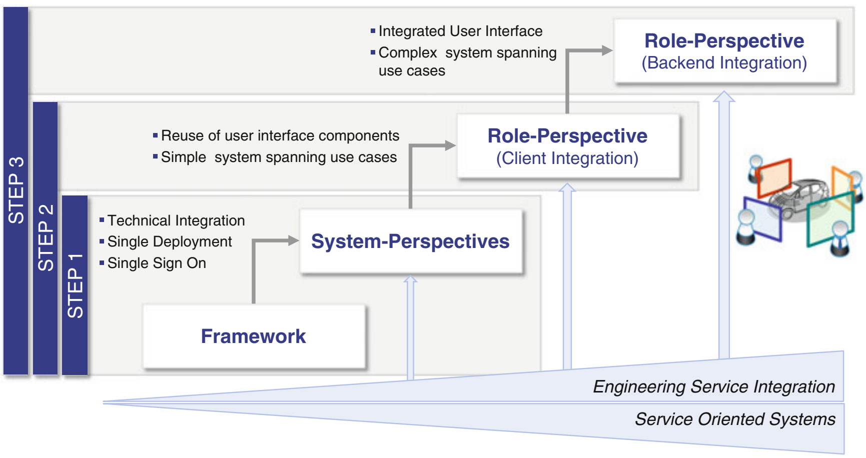


Fig. 21.9 Steps to develop a common client architecture [15]

21.4.5 Semantic Nets for Backend Integration

Imagine you are an engineer confronted with the task to redesign the middle console in the car. In order to perform your task appropriately you need to answer questions such as: why does the assembly need to be changed? What were the requirements for the change? Who had first designed it? Who is in charge now? In case you are a procurement manager whose task is to order new parts for production you might have similar questions. Your procurement process takes 12 months and you know that changes in the design process may have severe implications on the stock needed, particularly on cost but also on timely delivery of the products. Who should you ask for updates on the changes? What parts and quantities should be ordered instead?

Unfortunately, in both cases there may be several business applications storing relevant data that you require. Screening them, however, is a very laborious task and most probably not even sufficient to answer all your questions. Instead, you may likely need to consult experts whose names you don't even know. In short, you are confronted with a complex business world that lacks transparency and gives you the feeling of being lost in data space. Of course, the engineer and the procurement manager are only representatives of example user roles facing problems of similar complexity. There are many comparable roles. The reasons for these problems have already been mentioned throughout this chapter: Recent IT solutions are not designed to holistically and sustainably deal with increased product complexity, exploding number of variants, new demands for supplier integration, internationalization as well as M&A activities. In addition, there is no general system in sight that is able to cover all of these aspects across processes of a car manufacturer although it would be highly desirable.

Modularity, however, is a powerful feature that can be applied to IT in a more general fashion by applying it not only to data structures but also to the overall IT-landscape. The modules of a modular IT-landscape may in the first place range from complete software systems or components of these systems on the upper level down to business objects such as parts, assemblies, drawings, changes, etc. on a very granular level. The rationale behind is to leverage as many investments as possible made in IT infrastructure while at the same time obtaining a maximum of interoperability established between given the data systems.

The modular IT-landscape is in contrast to recent IT approaches which follow a classical top-down paradigm where the goal is to completely restructure a given landscape or part of a landscape to obtain a "clean world" over a defined period of time. However, this IT paradigm definitely has its limitations as many projects of this kind have failed, e.g., for the sheer size of the projects and involved cost, for lack of flexibility and for lack of keeping up with the high pace of modern car business. As a result the "new world" is already old when it is ready to be used but by then the investment is lost.

As a consequence of such experiences, it is helpful to accept the variety of solutions in software landscapes and not try to go against the tide. These solutions are usually driven by business needs. Variety should thus not generally be seen as a nightmare by IT responsible but as a chance if it is possible to horizontally manage these solutions, much in the sense of the modular IT-landscape proposed above. Thus, it combines bottom-up flexibility with centralized availability and provides room and freedom for quick adaptations to business needs.

A technical prerequisite for horizontal management is to establish connectivity—the glue relating data of different data worlds and keeping these relations up-to-date. This can be done by means of a semantic network that automatically links up the data of the business applications and thus provides the needed transparency. In this way it is possible to provide the engineer in the example above with exactly the information he needs even if relevant information is stored in remote business applications. Considering the chain of events starting at a decision to change the middle console that invokes a change process triggering a cascade of work orders which causes the engineer's demand may be represented as a logical deduction process of the semantic network (Fig. 21.10).

Once the network is in place, it allows the definition of user- and role-specific applications, e.g., an intelligent search engine allowing the connection of objects in distributed systems, traceability applications making the impacts of complex change processes transparent but also call back actions and many other APPs.

The semantic network approach is very well aligned with the SOA-based enterprise service bus (ESB) Architecture [19]. In fact, it extends the service bus with a Digital Brain, e.g., a part knows the assembly where it is part of and the assembly knows the change it was affected by and it also knows where it was built in and one could continue extending this chain of connections. This kind of connectivity information combined with the service bus allows for intelligent information logistics. Whenever a change takes place this may be known to all relevant players of the process (even beyond process borders) in real time. Individualized push-services may be designed as business rules to make information accessible and exchangeable. Even business intelligence may be empowered, on the one side through intelligent context information delivered by the network and on the other side by the fact that the information is available in real time and not through data loading processes preserving media gaps.

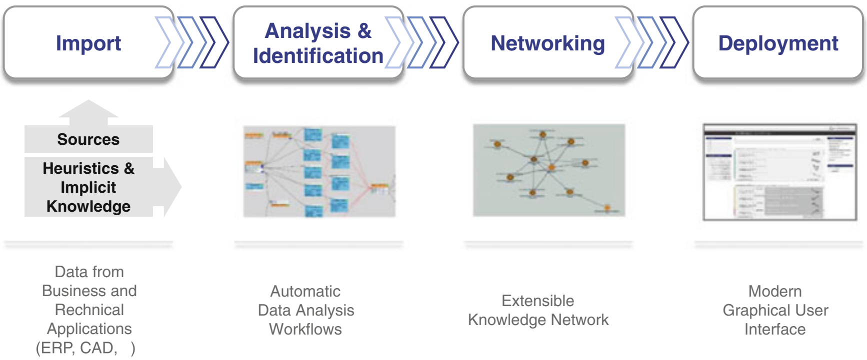


Fig. 21.10 Using semantic network approach

21.5 Concurrent Engineering in Automotive Development

Concurrent Engineering in the automotive industry has different aspects to be considered:

- A vehicle contains different modules which are developed in parallel but have to function comprehensively.
- Each module normally has mechanical parts, electronics and software which have to fit together to realize the required function.
- In case of in-house development external engineering service provider contributes to the product definition.
- Reflecting on Fig. 21.3 lots of components are developed and produced internationally.
- For international production global logistics have to be considered.
- In case of development and production of local product extensions specific modules have to be modified or developed in parallel.

Figure 21.2 shows the tremendous expansion of vehicle variances over the last years. This has been realized more or less without increasing the number of engineers working on the OEM side. In the past, an engineer had a defined task in a vehicle program dealing with a limited number of suppliers on the next tier levels.

Today the same engineer has to take care about different vehicle programs, dealing with a significant number of partners on different tier levels, and take into account the requirements of other OEM's in a partnership. Depending on the single project he has to play different specific roles. Even the relationship OEM ⇔ Supplier can change case by case and plays case by case specific roles [21]. For example in an OEM ⇔ OEM cooperation it is possible that OEM 1 provides small sized engines to OEM 2 and is so in the supplier role while OEM 2 provides bigger sized engines to OEM 1 as supplier. In both cases potentially the same groups of engineers have to cooperate.

To tame the increasing complexity, the role of system integrators becomes more and more important. Figure 21.11 displays this tendency.

To find a way to confirm relationships in such a network the PLM working group of "German Automotive Association" (VDA-AK PLM) published the "VDA Recommendation to harmonize data logistics in simultaneous projects" (VDA 4961/3, released in April 2012) [22]. The purpose of this recommendation is to harmonize the CE organization and data logistics within the framework of collaboration in **S**imultaneous **E**ngineering **P**rojects (development partnerships).

The VDA recommendation assists the preparation and conduct of the project work by:

- Classification of the cooperation models to differentiate the development partnership. The cooperation models classify the development partners according to their role in the development partnership and assign specific characteristics to them.
- Using the Simultanous Engineering checklist and the corresponding templates to harmonize the data logistics in SE projects with regard to the predetermined

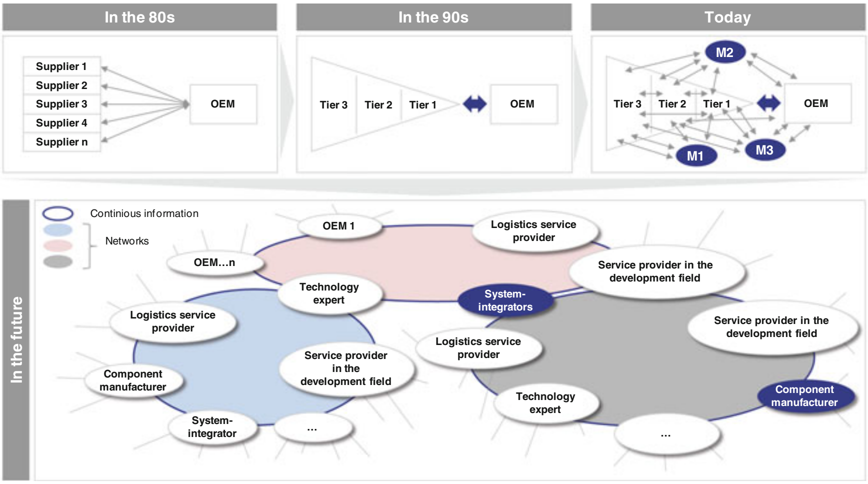


Fig. 21.11 Collaboration networks in vehicle development [20]

cooperation models. The following topics are dealt with using the SE checklist and the corresponding templates:

- Specific communication and CA infrastructure of the project partners
- Determination of data exchange formats for technical documents (data exchange contents, quality and method)
- Process-oriented stipulations
- Project management, change and release procedures
- Determination of deadlines, times, costs and responsibilities
- Legal aspects.

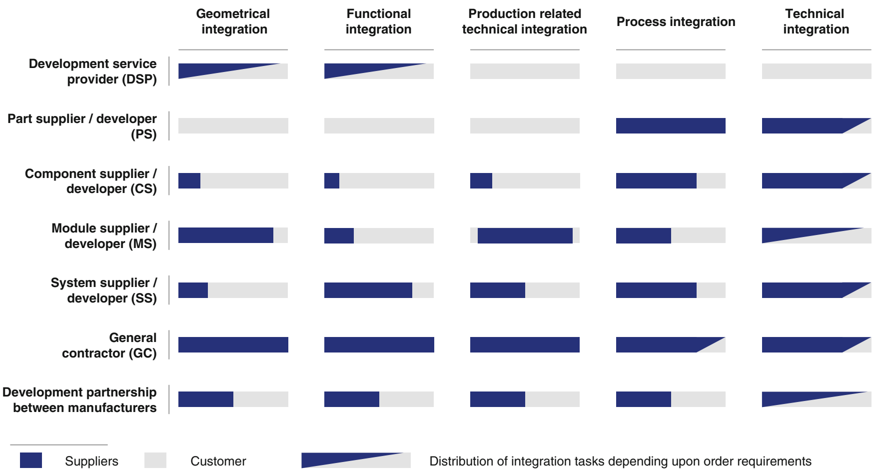


Fig. 21.12 Definition of collaboration roles in VDA 4961/3 [22]

The recommendation is clustered into 4 parts:

- Introduction to define objectives, target groups and VDA activities
- Definition of different roles (Fig. 21.12)
- Criteria matrix for cooperation models
- Process model for implementation of a cooperation.

21.6 International Standards

In the 80s a large number of different CAD systems were developed. This made the necessity for standards for data exchange obvious. At the beginning, dedicated standards for drawings and 3D models appeared. In 1984 the development of STEP started as a successor of formats such as IGES, SET and VDA-FS (Fig. 21.13). The initial plan was to develop one implementation independent product data model for all purposes. Because of the complexity the standard was modularized. In 1995 ISO published the first initial release. Based on these modules different user communities could create application protocols that adapt the data model and definitions to the specific requirements of the business. For automotive AP 214 was defined and enhanced continuously [23].

With the upcoming internet technology new standards for XML schemata and Web services appeared. With the new AP242 a consolidation of different application protocols is going to be done and updated with new internet technologies.

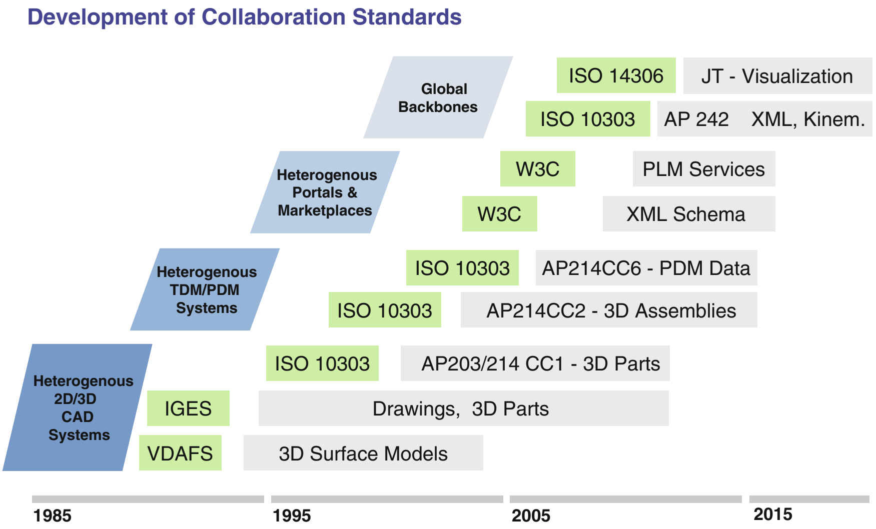


Fig. 21.13 Continuous development of cooperation standards [23]

Beside CAD usage visualization of 3D data became more and more relevant. So a broader community of people involved in product development could leverage from 3D data but with a significant reduced size. In this area the JT-format became more and more importance. So it was a major interest for automotive industry to standardize this format. On December 2012 JT became ISO 14306.

21.6.1 Code of PLM Openness (CPO)

For commercial systems a kind of compliance check has to be developed which helps to evaluate potential solutions on their integration abilities. The fulfillment of this compliance check is a mandatory aspect for selecting the commercial solution.

To get a better overview of the capability of the solutions provided by the market the automotive industry has defined under the leadership of ProSTEP iViP Association, a catalog of criteria of openness of systems and service solutions [24]. The motivation to develop such a common document was to get a common view on:

- Standardized user-interfaces for different systems
- Compatibility of functions provided in different systems
- Subsequent use of upstream data
- Mastering complexity of IT landscapes
- Mastering IT operations
- Mastering data consistency and correctness

The aim of the CPO is to get transparency on openness of PLM software by impartial, non-discrimination criteria graduated in “shall”, “should” and “may” for criteria described in Fig. 21.14.

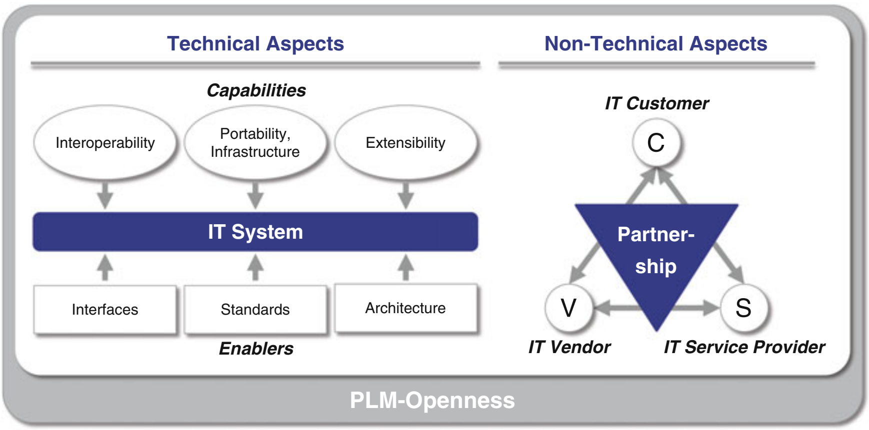


Fig. 21.14 Criteria of “code of PLM openness”

The intent is a voluntary self-commitment where it is possible for a vendor to document his fulfillment of CPO criteria in a public statement.

The CPO community is open for everybody. In the first year of publication of CPO in March 2012, more than 80 international companies joined the community. Like the first definition, a voluntary group of specialists are working on the enhancements of the paper as well as on implementation guidelines and recommendations for adaption.

In a relationship between a user company and an IT vendor, CPO can give a guideline for defining a contract. Here the general statements of CPO can be detailed case by case in the specification of the cooperation and deliverables.

The defined "common language" of system openness is going to help everybody in the community to get a better understanding and communication.

From the perspective of automotive companies, CPO will leverage the following advantages:

- Sustainable reduction of IT effort for
  - Integration of new application and systems
  - Release—and system change
  - Operation and maintenance effort
- Simplification of collaboration between system provider, IT-consultants and user
  - Common understanding and common semantics about PLM openness
  - Rules and commitments for implementation
  - Regulations for contractual decisions
- Optimization of system usage in business processes
  - Utilization of best in class solutions from different vendors
  - Acceleration of process—changes and optimization.

21.6.2 ISO 10303 (STEP AP 242)

The STEP application protocol 10303-242 "Managed Model Based 3D Engineering" represents a logical extension to the use of STEP for product data exchange and product data integration in order to meet the demands of today's manufacturing industry and to keep pace with progress in the field of information technologies [25].

As part of an efficient standardization strategy, the two key standards for the automotive and aviation industries, "ISO 10303-203—Configuration Controlled Design of Mechanical Parts and Assemblies" and "ISO 10303-214—Core Data for Automotive Mechanical Design Processes", were combined to create AP 242 and new concepts were added. On the one hand, this will safeguard the investments made in the standardization and implementation of STEP in recent years, in particular with regard to the exchange of CAD data. On the other, the addition of new functions will satisfy the demands of the automotive, aviation, aerospace, and defense industries which act as the principal drivers and sponsors of this activity.

The most important new functions are:

- Composite materials,
- Kinematics,
- Enhanced product manufacturing information (PMI),
- Data quality in geometric models.

AP 242 also marks the introduction of a new architectural concept into STEP: the Business Object Model. This represents the most important concepts and information of a subset of AP 242 using the terminology employed by specialists from the aviation, aeronautics, automotive and defense industries. To summarize, the Business Object Model covers the non-geometrical functionalities of AP 242 such as product structure, document structure, meta-data, kinematics, and composite materials. The Business Object Model forms the basis for an XML schema. This standardized XML schema supplies the context for geometrical information that can be represented in other formats (JT, native, PDF, etc.). This means that the geometry related data in an AP 242 XML structure is mapped to external formats by means of references. In addition, AP 242 also contains a new external element reference mechanism. In the past, it was only possible to reference external files that contained a geometrical model (e.g. from the assembly structure through to the individual parts). Thanks to the new functionality, it is now also possible to reference elements in the external file (e.g. a curve, edge, axis). The external file may be of any type (e.g. JT, native, or STEP itself). This is extremely important, including for kinematics and PMI at assembly level. At the kinematic level, for example, it may be necessary to be able to describe the structure in STEP and the geometry of an articulation in JT.

AP 242 permits the long-term safeguarding of the investments made by industry in data exchange and integration: It is upwardly compatible with the existing STEP interfaces available in CAD systems. It also permits interoperability with other formats such as JT (Fig. 21.15).

AP 242 is more than just a reengineered version of functionalities familiar from existing standards such as STEP AP 214 or AP 203. It offers a genuine value addition because it provides a response to the new requirements arising from modern production technologies and engineering methods.

As aforementioned, AP 242 is intended to become the common international standard for the manufacturing industry, across the various domains and sectors. The aim is to provide a robust, powerful foundation for constructing sector and enterprise-specific solutions. The modular nature of the standard means that industry as whole, individual sectors, individual enterprises or individual users can select the information that is tailored to their specific requirements. This can then form the basis for constructing context-specific information or other services. The modular approach also means that AP 242 can be introduced more quickly by industry.

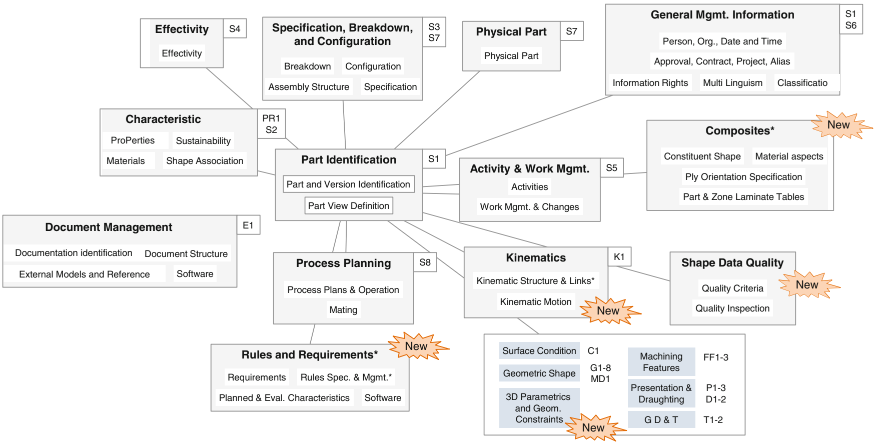


Fig. 21.15 Planned data model of ISO 10303-242 [25]

A common cross-industry standard also has advantages for system suppliers. Globalization is resulting in the same tools being used worldwide, irrespective of whether the product being developed is a car, a plane or a satellite.

AP 242 covers the entire development process from the early planning phases through to operation. Because AP 242 has been harmonized with existing solutions for product lifecycle support, this transition will not result in any discontinuity in information provision. The following list of use cases is not exhaustive. It is simply intended to indicate the wide range of applications of AP 242:

- Replacing technical drawings: Engineering design using a 3D master model containing all the information that is usually found in a technical drawing, e.g. PMI (Product and Manufacturing Information).
- Long-term archiving.
- Viewing: Structure, geometry and metadata of a product are displayed in a viewer in order to make it possible to check the dimensions, shape, adjacent parts, etc. Product data and non-geometrical metadata is represented in AP 242. CAD and visualization data is represented in other formats that can be referenced from within an AP 242 depiction. Other formats are more suitable than STEP for the lightweight exchange of geometry data, in particular for viewing purposes. However, it is only in combination with AP 242 as the process backbone that the full benefits of this type of application can be exploited.
- Multi-body simulation: AP 242 makes it possible to describe kinematic structures (links, articulations, pairings, movements, etc.). These are extended by geometric models such as JT, for instance. Links from the kinematic structure to the geometric elements are established via external references.
- Maintenance: Lifecycle support is not the primary aim of AP 242. Despite this, AP 242 provides the product development and engineering-related information about a product that is required to make lifecycle management possible. To this

end, AP 242 has been harmonized with the STEP standard on product lifecycle support.

One of the main feature in the new STEP extension is its deep interoperability with ISO 14306 (JT) which covers the recently standardized concept of digital engineering visualization (see Chap. 11) based on experiences in various industries [26]. JT covers capabilities which fulfill the demands for visualization and data exchange of complex products like car or aeroplane in an uniformed way. Although JT can be used standalone with corresponding tools, as practiced successfully in the past decade, it becomes even more powerful in combination with STEP AP 242, in particular in case of supplier integration which is described in the following Sect. 21.7.

21.6.3 ISO 10303 and ISO 14306: A Powerful Combination

JT and STEP AP 242 are on the way to change CAD based development and gain much more benefits out of 3D data than possible in the past.

The ISO standards provide a reliable and sustainable definition for product describing data including product structure and behaviour models.

Powerful low cost or freeware solutions will appear for a more intensive use of product data. Special solutions for ultra-light weight descriptions under JT definitions are on the way to be established.

For DMU and other integration processes international and multi partner collaboration can be defined in a new way. Here the specific solutions of each partner can be respected as long as he is committed to the two ISO standards [27].

Design in context solutions have a better chance when they support JT and AP 242. This will enforce the reincarnation of the special CAD solutions for dedicated tasks. Secondly special solutions will be developed directly using JT definition. The one fits all CAD “Dinos” might have a tough future [28].

21.7 Supplier Integration

The value addition in many industries is decreasing on OEM side and increasing on supplier side (Fig. 21.3) [29]. In the automotive industry suppliers often deliver complete car modules such as door panels or dash boards. Therefore, supplier integration is a highly challenging task, on the one hand, and promising substantial savings when executed efficiently, on the other hand. As it belong almost the entire functions of an enterprise [30], further explanations in this section will be focused on the field of the engineering collaboration (see Chap. 7).

The integration concepts depend primarily on collaboration roles, as highlighted in Fig. 21.12 [22]. In case of deep collaboration (development partnership, general contractor, system supplier) a performant PDM-PDM exchange or interoperability is mandatory and already implemented in many cases (see Chap. 16). This exchange encompasses workflows, structures and all related documents which are exchanged or shared by a pre-defined process frequency which ensures that the related contents can be kept up-to-date. This concept is highly challenging for both parties, expects clearly defined processes and operational excellence. The underlied software solutions are usually powerful and expensive.

As a derivation of this concept, most OEMs allow their suppliers to exchange the CAD and PDM data by using smart PDM clients. Apart of costs for a performant web connection and efforts for manual handling of exchanged data, this concept gives a clear interface with well defined, limited functionality which allows the definition of clear responsibilities between the parties. Nevertheless, this concept has been seen not entirely satisfying the needs of many suppliers which have no demand for continuous exchange on daily basis. In this case suppliers can employ a service provider which conducts the data handling, exchange, translation, and adaption. This concept is promising if it can be used to perform the customer process to all OEMs [31]. It requires no invest, defines a clear interface to customers and causes costs per use only.

In case of module, component and part suppliers, the deep PDM integration is not required. For that reason, a data exchange with product structure, related CAD models and small extent of PDM data is a feasible way to fulfill needs of both involved parties. In such a concept, a data package is exchanged between parties. The geometry data can be exchanged as native data or neutral files (STEP, JT). Product structure and attributes of assemblies and parts are stored in a PDM header file (PLM XML or STEP). While system landscapes in context of global sourcing processes are typically heterogeneous the usage of ISO standardized data formats (JT/STEP AP242) for data exchange is getting more and more important. Such STEP assembly manager (SAM) is capable to exchange not only entire structures in desired approval status in bidirectional way, but the limited extent of initial package too which is subject of engineering change. Figure 21.16 gives an overview of the process landscape in such supplier integration scenarios. These processes are

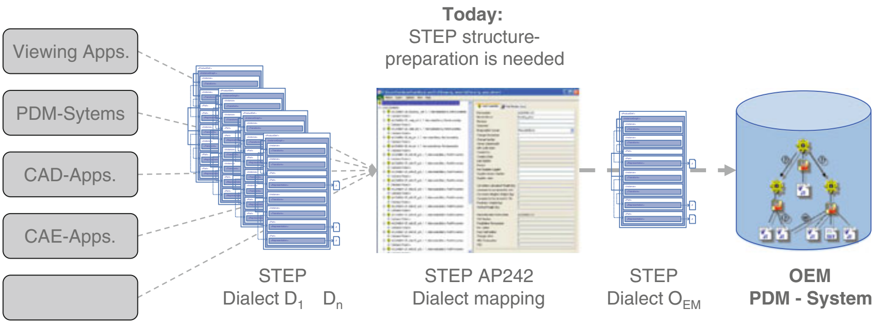


Fig. 21.16 Structure data exchange in collaboration context

validated in automotive industry in several pilot projects e.g. with global automotive suppliers.

The results of these JT-based data exchange processes are very promising and further potentials in using JT instead of native 3D-CAD data in collaboration processes are identified e.g. in the fields of data preparation, handling and storage. To reach such a duo of complemented and ISO standardized data formats a lot of work has to be done [28].

21.8 Knowledge-Based Engineering

“Design Templates” as knowledge-based applications are a comprehensive approach for archiving and managing all essential information in a standardized product and process description (see Sect. 10.4).

Each car line, each assembly, each component contains various and numerous artifacts that influence dedicated development steps. Starting with the conceptual design, crossing all design stages and ending with data archiving, sophisticated development methods and IT solutions must be ensured [32]. Seamless and just-in-time information providing all downstream processes and an unambiguous and easy performable process definition are assumed [33]. Using template technologies is the key to handling most of these aspects in a modern CAD system. The schema in Fig. 21.17 shows the different kinds of templates structured by the level of geometrical completeness [34].

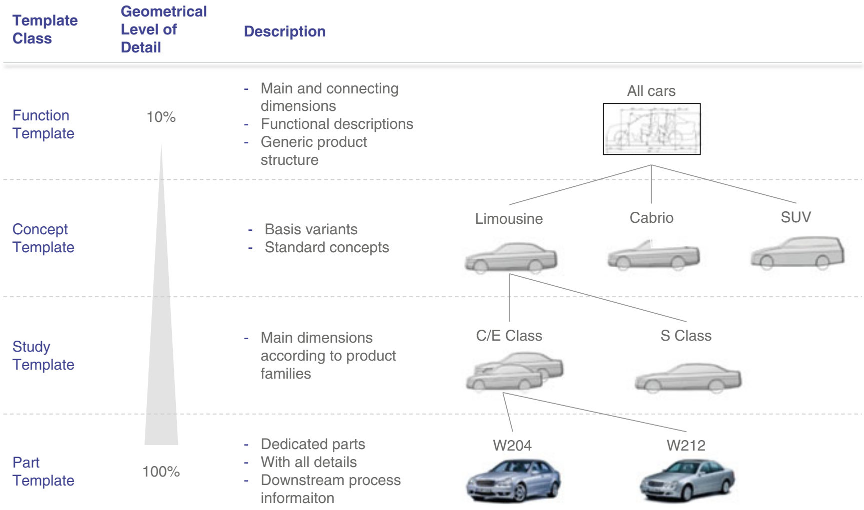


Fig. 21.17 Classification of design templates in the automotive development

Application of concept templates includes main characteristics of vehicle models such as sedan, convertible, station wagon, or SUV. They are the foundation for best practice design concepts. The digital validation of functional principles is the task of study templates. The detailing of such a validated concept leading to a full geometrical description of parts, including relevant information for manufacturing and final assembly, can be done in part templates. Within all layers, the design engineer can use specific templates for the different modules of a vehicle.

To provide the opportunity to include all geometrical and non-geometrical information independent of the process step, a specific PDM archiving concept has to be developed. It enables data retrieval with different points of view. A generic information structure, independent of the level of detail, is the base of the archiving templates.

This structure is a summary of different information aspects of a comprehensive product description. Depending on the concrete development task, the necessary information is activated and shown in the expected context. The structure distinguishes between parts with product part number and so-called arrangement (support elements). The generic information set creates the structure for all input data for the templates and links all underlying datasets existing in the PDM database to the part description.

Only a suitable PDM solution can ensure such a dynamic information flow. The sophistication of CAD functions requires a higher level of PDM capability [35]. A real valuable benefit can be achieved only through the integration of CAD and PDM. In addition to the known PDM requirements, such as configuration management, versioning, and release and change management, the capability to administer constraints are essential. This means especially the constraints between geometrical elements and parameters within parts as well as constraints between parts and subassemblies. More than 2,500 links are needed to define an entire body-in-white structure within a concept template. This link management gives the capability of dividing complex structures into template based and usable part structures. Without this capability, it would be impossible to share the complete information and knowledge of a multi-part assembly among numerous design engineers. The mandatory use of template-based design processes leads to a continuous improvement of the design maturity, from the early phase down to detail design, and prevents countless iteration. The reuse of these approaches depends on the degree of flexibility and adaptability of the predefined templates. The predefinition, by using knowledge-based form and function features, facilitates this reusability. These feature applications are not only part of detail design, they can also define and mutate conceptual structures through an internal protection structure.

Using knowledge-based templates is an appropriate approach for integrating proven concepts or systems into a new product design [36]. They contain all the information necessary to define the technical behavior in a general context. The disadvantage of this approach is obviously the intensive effort needed to define and maintain a universal template concept that considers all potential variants of future design instances.

To succeed in the development and deployment of such a sophisticated concept, technical and conceptual aspects must be considered. The most important part of the game is the human being—the engineers and designers who have to perform this new process and methods.

21.9 Conclusion and Outlook

In the last decades from the early beginning of engineering IT until the millennium, the IT-vendors were in a concentration phase. Lots of mergers and acquisitions were done. In parallel the user side also started with a variety of systems and ended up mostly in a monolithic approach. These systems are developed and customized with a clear focus on OEM internal process demands, but are mostly inflexible and scalable only with high effort.

Given the current and future complexity of product development with the huge product differentiation, the increasing number of partners involved and the tendency of internationalization of product development and production, this monolithic system strategy inside automotive OEMs is not sufficient anymore. On the side of partners each supplier was forced to run the monolithic system environment of each OEM.

The upcoming demand on system engineering will bring in some more process demands which can only be partially fulfilled by the current solutions. Today it is a competitive advantage to establish a collaborative process environment between two partners (OEM ⇔ OEM or OEM ⇔ Supplier) in a short time. However the monolithic systems in place today prevent a fast setup of a collaborative environment.

To overcome this situation, a more modularized system environment is required. But this cannot be achieved in one big bang; it must be realized in a step by step approach. SOA are the key technologies for this journey. However it is not only a technical or commercial question. It is also an organizational change, on the user side as well as on the side of IT responsible.

Initiatives such as “CPO” and standards like ISO 10303-242 and ISO 14306 are the key enablers.

The process development of systems engineering with a clear methodical orientation along a requirements, functions, logics, and physics (RFLP)-approach will bring in additional demands. A continuous development of systems and methods based on SOA and international standards will help establishing an engineering IT world, where concurrent engineering spanning over different domains, companies, and regions should be possible with the same efficiency as inside each company.

References

1. Genta G, Morello L, Cavallino F, Filtri L (2014) The motor car past, present and future. Springer Science + Business Media, Dordrecht
2. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manage 7(2):101–115
3. Vondruska J (2012) MQB in-depth: a look inside the golf7 platform. http://www.vwvortex.com/features/technical-features/vwvortex-feature-a-first-look-at-the-new-golf-7-mqb-architecture/. Accessed 15 Mar 2014
4. Riedel O (2010) Collaborative Engineering – steht ein Generationswechsel an? 5. Forum Innovationscluster „Digitale Produktion“, Stuttgart, 9 Dec 2010
5. Reinertsen D (1983) The search for new product killers. Electron Bus 9(7):34–39
6. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manage 6(4):307–323
7. NN (2012) Global research highlights. Bank of America Merrill Lynch, New York
8. NN (2012) FAST2025—Future automotive industry structure, vol 45. VDA, Berlin. http://www.vda.de/de/publikationen/publikationen_downloads/detail.php?id=1121. Accessed 15 Mar 2014
9. NN (2014) Library of PMI global standards. Project Management Institute, Newtown Square. http://www.pmi.org/en/PMBOK-Guide-and-Standards/Standards-Library-of-PMI-Global-Standards.aspx. Accessed 15 Mar 2014
10. Balasubramarian B (2008) Entwicklungsprozess für Kraftfahrzeuge unter den Einflüssen der Globalisierung und Lokalisierung. In: Schindler V, Sievers I (eds) Forschung für das Auto von morgen. Springer, Berlin, pp 359–372
11. Planing P (2014) Innovation acceptance: the case of advanced driver-assistance systems. Springer Fachmedien, Wiesbaden
12. Maurer M, Winner H (2013) Automotive systems engineering. Springer, Berlin
13. Katzenbach A (2011) Strategic development of daimler engineering IT. EDM CAE Forum. Stuttgart, 12.07.2011
14. Srinivasan V (2011) An integration framework for product lifecycle management. Comput Aided Des 43(2011):464–478
15. Katzenbach A, Riemann R (2009) Next generation engineering IT: transformation der IT-Landschaft bei Daimler. SOA Days, Bonn, 25–26 Mar 2009
16. IT Governance Institute (2005) Governance of the extended enterprise. Wiley, Hoboken
17. Katzenbach A, Steiert HP (2011) Engineering IT in der Automobilindustrie – Wege in die Zukunft, Informatikspektrum 34(1):7–19
18. Katzenbach A (2010) Engineering IT heute – Wege in die Zukunft. Arbeitskreis Softwaretechnologie der Region Bodensee, Konstanz, 12.11.2010, http://akswt.uni-konstanz.de/docs/2010-04_Konstanz.pdf. Accessed 15 Mar 2014
19. Kamps T (2013) Komplexität in automotive PDM/PLM transparent machen durch die Automatisierung von Wissensprozessen in der Produktentwicklung, ProSTEP iVip symposium, Hannover, 16–17 Apr 2013
20. Göschel B (2006) VDA Technischer Kongress, München, 22–23 Mar 2006
21. Jäger AL (2014) Global purchasing processes in the business sector automotive aftermarket development of a reference model. Springer Fachmedien, Wiesbaden
22. NN (2012) VDA Guideline 4961/3: recommendation to harmonize data logistics in SE projects, VDA, Berlin
23. Katzenbach A (2013) Lecture Informationstechnik und Wissensverarbeitung in der Produktentwicklung, University Stuttgart
24. NN (2010) Code of PLM openess. ProSTEP iVip Association, Darmstadt, http://www.prostep.org/en/cpo.html. Accessed 15 Mar 2014
25. Ungerer M (2013) Managed model based 3D engineering. ProSTEP iViP Association. Production Data Journal Nr 2 :20

26. Katzenbach A, Handschuh S, Vettermann S (2013) JT Format (ISO 14306) and AP 242 (ISO 10303), The step to the next generation collaborative product creation. In: Kovacs GL, Kochan D (eds) Digital product and process development systems. Proceedings of the IFIP TC5 international conference NEW PROLAMAT. Springer, Heidelberg, pp 41–52
27. Handschuh S (2011) Wertextrahierende Nutzung von offenen leichtgewichtigen Datenformaten in automobilen Kollaborations- und Entwicklungsprozessketten, PhD thesis, University Kaiserslautern
28. NN (2010) Applying JT, whitepaper. ProSTEP iViP Association, Darmstadt. http://www.prostep.org/en/medialibrary/publications/white-paper-studies.html. Accessed 15 Mar 2014
29. Hundertmark H (2013) Beziehungsmanagement in der Automobilindustrie: OEM relationship management als Sonderfall des CRM. Springer Fachmedien, Wiesbaden
30. Senescu RR, Haymaker JR (2013) Evaluating and improving the effectiveness and efficiency of design process communication. Adv Eng Inform 27:299–313
31. Bondar S, Potjewijd L, Stjepandić J (2012) Globalized OEM and Tier-1 processes at SKF. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering, Springer, London, pp 789–800
32. Bodein Y, Rose B, Caillaud E (2013) A roadmap for parametric CAD efficiency in the automotive industry. Comput Aided Des 45(2013):1198–1214
33. Park YW, Fujimoto T, Hong P (2012) Product architecture, organizational capabilities and IT integration for competitive advantage. Int J Inf Manage 32(2012):479–488
34. Katzenbach A, Rolinger A, Bergholz W (2007) Knowledge-based design—an integrated approach, 17th cirp design conference future of product development. Springer, Berlin, pp 13–22
35. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manage 7(3/4):324–347
36. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manage 7(2):116–131

Chapter 22
Concurrent Engineering in Machinery

Jožef Duhovnik and Jože Tavčar

Abstract Application of concurrent engineering (CE) to machinery has to consider the type of production (individual, serial), product complexity and level of design. Product development (PD) involves four characteristic levels of design that requires specific activities. The characteristic design levels require definitions of the activities for providing the necessary software and other support for all phases of the design process. The following four levels of the design process have become established in the professional literature: original, innovative, variation and adaptive. Systematic analyses of product development processes (PDP), workflows, data and project management, in various companies, has shown that specific criteria have to be fulfilled for CE to be managed well. It is very important to consider the involvement of customers and suppliers, communication, team formation, process definition, organisation, and information system to fulfil minimum threshold criteria. The quality of communication and team formation, for example, primarily affects the conceptual phase. An information system is useful predominantly in the second half of the design process. It is shown with typical examples what is important in each PD phase. In the second part of this chapter reference models for CE methods are presented for PD in individual production (CE—DIP), in serial production of modules or elements (CE—DSPME) and in the manufacture of mass products (CE—DMMP) with an example from household appliances. The reference models for CE methods map PD phases and CE criteria for each type of production and have to be used together with case studies. They help to recognise strong and weak points of a CE application and show a way to improve processes and supporting CE methods.

Keywords Product development process · Golden design loop · Reference models · Design levels · Product complexity · Concurrent engineering criteria

---

J. Duhovnik (✉) · J. Tavčar
Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia
e-mail: joze.duhovnik@lecad.fs.uni-lj.si

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_22

22.1 Introduction

Product development (PD) involves four characteristic levels of design [1]. Each of them requires specific activities (Fig. 22.1). These characteristic design levels therefore require definitions of the typical activities in different product life cycle phases. On the basis of the above design levels, design tasks can be specified and assigned to them. Better understanding the needs of the various activities enables the creation of better software and other support for the different phases of the design process. There are four levels of the design process: original, innovative, variation and adaptive.

Original design means the design of entirely new products, whereby a new working principle is determined for a new or known function. In the process of designing from scratch, one therefore needs to define the working principle, model of shape, functionality and technical shape.

Innovative design means the design of products by varying the working principles which fulfil the required function to an optimum degree. In innovative design one needs to define the model of shape, functionality and technical shape.

Variation design means the design of products for different loads or capacity levels. At variant design products have comparable models of shape. In variation design one needs to define the (added) functionality and technical shape.

Adaptive design means the design of products by adapting their dimensions to the technical and technological possibilities for their manufacture. In adaptive design one needs to define the technical shape. This shape is conditioned both by optimization of micro-technology (special features of the manufacturing technology) and by the shape design of details (ergonomics, assembly, etc.). Adaptive design is a dominant type of design and typical of the engineering change process.

Characteristic design phases are: design planning, conceptual phase, system level design, detail design, testing and refinement, production ramp up [2]. During

	PRODUCT LEVELS	DESCRIPTION	DESIGN LEVELS	% OF ACTIVITIES
I.	New product / new technology	New working principles	Original Design	0.2 … 0.8%
II.	New product / known technology	Variation in working principles	Innovative Design	2 … 5%
III.	Variant of an established product	Variation in Functionality	Variation Design	10 … 25%
IV.	Product improvement (change)	Variation in Dimensions	Adaptive Design	75 … 85%

Fig. 22.1 Relationships between design and product levels [1]

their work, designers will require different types of support, depending on the phase of design or abstraction of the product they are working on [3–5].

A product's complexity (Fig. 22.2) has an important influence on the product development process (PDP), presented from different points of view. Individual stages in PDP should be optimized in accordance with the type of the production process and the product's complexity. Products have different types of complexity (Fig. 22.2) [6]. A complex product is seen as a comprehensive product from its design point of view, e.g., a camera mechanism. Simple products in mass production are often very complicated from the technological point of view (e.g., electric bulb). The adjustment of products to customers often results in a vast number of variants, which are difficult to manage (e.g., household appliances). Products composed of a vast number of different elements are also complex. Each type of complexity requires a sound arrangement of selected phases in PDP (Fig. 22.3). For example, a product that is complicated from the construction and technological points of view requires a detailed preparation while products, composed of many elements are complicated from the view-point of logistics of supply chain and production. The PDP, described on the basis of some characteristic examples of product's complexity and volume of production, is presented (Fig. 22.2).

A product is defined with basic data as function, shape and its structure. Material is a dependent parameter, which defines volume and capability of the product in different environments. If the complexity of a product is increasing from a simple mechanical element or structure to a complex system with parameters: function, structure, shape, material, and organisation, all those five parameters define the product. Interaction between the five parameters is happening inside the golden loop as explained in the next section (Fig. 22.3).

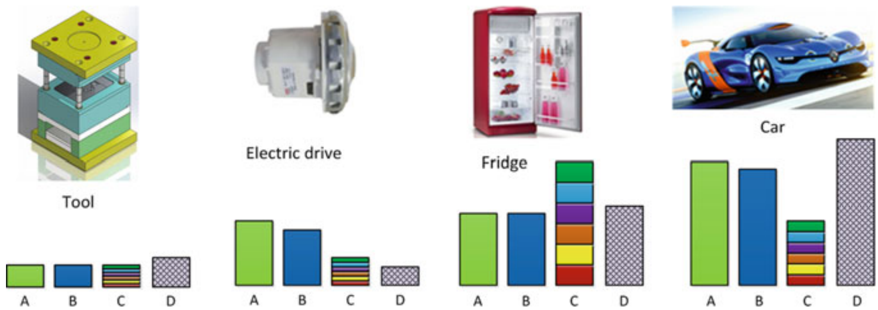


Fig. 22.2 Examples of different types of product's complexity. Complexity of the product: *A* Complexity of the product's design. *B* Complexity of the product's manufacturing processes. *C* Number of variants. *D* Number of parts

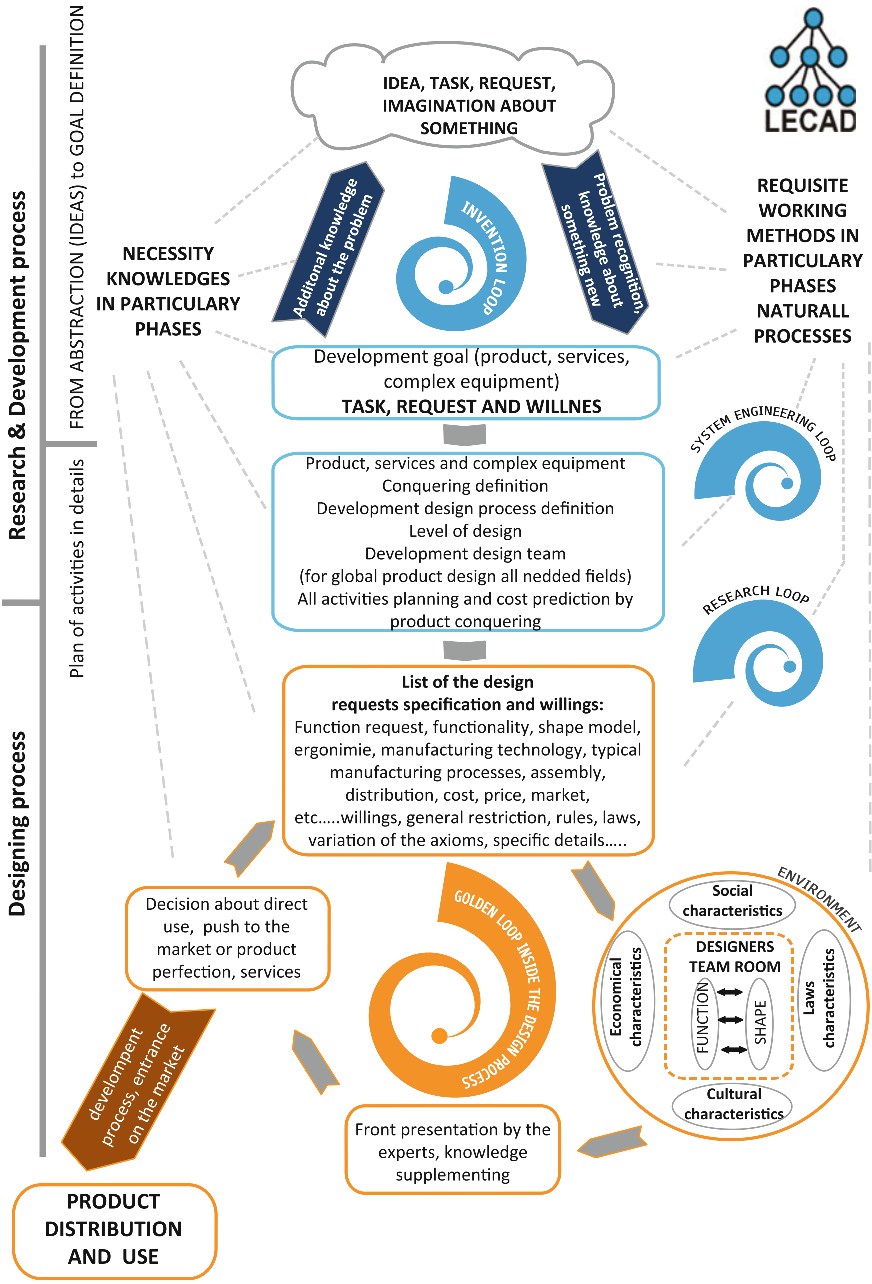


Fig. 22.3 General PDP

22.2 Generalized Model of a PDP

The generalized model of PDP (Fig. 22.3) helps us understand and compare procedures in different types of production and, thus, find the most appropriate methods for an enterprise. New PD begins with an idea. In the first invention loop specification needs to be transformed into the development goal, a first version of product specification [7]. Stimulating employees to be creative and collection of proposals for new products must be enabled throughout the company and also from outside the company, service personnel and salesmen being the most important participants. New product ideas collection needs to be supported with idea assessment and company strategic development planning.

In the planning phase that includes systems engineering and research the new product idea should be transformed into a project definition. Product design is finally executed inside the golden loop. Product level and complexity define how dominant particular design phases are. At the variation and adaptive design level most of the activities are executed inside the golden loop. At the original and innovative design level the research loop is much more important. For research of working principles specific tools like the Matrix of function and functionality can be helpful [4, 7].

CE principles are included in several iterations or loops [7]. When the conceptual design is created within the golden loop it is checked several times through all kinds of criteria. If there is a decision at an assessment point that a product design is not satisfying market requests and specification the design iteration is stated again. The generalized model of PDP helps us understand and compare procedures in different types of production and, consequently, find the most appropriate methods for a specific enterprise. Each phase requests specific support [8].

In the process we should start with identification of needs. The interdisciplinary team, which has members from different areas depending on the product and used technology, prepares the description of needs for a clear definition of requirements [9]. Some researchers propose needs, others propose requirements as a starting point. We believe that needs from consumers, market, or nature are the main trigger for any new design. Needs have usually not been articulated clearly and for that reason the interdisciplinary team has the task to articulate needs as a first level of requirements. If during the R&D and design processes some new requirements are recognized, the same interdisciplinary team could redefine them in the next phase. This means, that requirements are collected during the whole new PDP. For that reason the research and development process invades the whole process at all. The process is stopped when goals are achieved.

The objective of this chapter is to develop a method that helps production companies recognize weak points in their PDP and improve their approach of CE. Analyses in various companies have showed that specific criteria have to be fulfilled for PD to be managed well. The impact of an individual criterion depends on the type of production. A clear definition of the process and its organization is important for all PD phases.

22.3 Concurrent Engineering Methods for Product Development

All phases of the product lifecycle need to be taken into account at the planning and conceptual design phases, while also the entire product family and possibilities for upgrading have already to be envisaged. Expected results are better products and a reduced number of changes in later phases of PD and, especially, after production launch [10]. An appropriate assessment of CE methods is important for the CE improvement process. Lawson and Karandikart [11] has recognised six criteria that identify a number of common barriers to implementing CE: knowledge of CE, measurement of current and identification of future states, provision of a scoring system, ease of application, involvement of relevant people, identification of key phases for change. Ainscough et al. [12] suggest the combination of self-assessment and an implementation workbook for organisations to manage the change towards CE. Self-assessment includes PDP, teamwork, IT, supply chain and project management. Hrzek et al. [13] recognises several key performance indicators for continuous improvement of business processes: information by documents, degree of communication, degree of process description and process stability. Other authors [14] are measuring benefits derived from using the CE methods. Some of the presented models are general [11], others have specific application in construction engineering [14] and have, therefore, limited value for machinery.

Seven key areas that define the level of CE in PDP were recognised. CE models from the literature [10] were compared with specific requests in the machinery sector and known CE assessment models [11, 12, 14]. The authors have tested and supplemented the CE criteria during several product lifecycle management (PLM) application projects and process analyses. The CE criteria or methods are presented below, first in general form, later more specifically for different kinds of production processes. Our key criteria or methods for CE are:

1. Interaction with customers (sales, distribution)
2. Involvement of suppliers (supply chain)
3. Communication (human interaction)
4. Team formation (different skill, all skills involved)
5. Process definition (workflow)
6. Organisation (soft organisation)
7. Information system (interoperability, dynamic structures).

Each CE criterion (Fig. 22.4) has five maturity levels for assessment, which have been defined for each CE criterion section. A higher maturity criteria level includes lower levels in general. Some maturity levels can be treated independently from others. This means that implementation of a higher-level CE criterion does not assure the adoption of a lower level. CE criteria and maturity levels are used later in our reference models for CE methods for different types of productions. The criteria and their maturity levels will be explained in the following paragraphs.

1 Interaction with customers	2 Involvement of suppliers
1.1 Written specification of customer's requests	2.1 Suppliers are monitored according to APQP (Advanced Product Quality Planning)
1.2 PD team participation at collecting of customer's requests	2.2 Suppliers are selected as long term strategic partners
1.3 Customer is involved at product validation	2.3 Established information connection for document exchange (PLM)
1.4 Customer is involved at project mile stones checking	2.4 Active participation of suppliers in conceptual and later phases of PDP
1.5 Active involvement of customer at product development	2.5 Systematic long term development of suppliers
3 Communication	**4 Team formation**
3.1 PD core team is collocated in one office	4.1 Multidisciplinary PD core team with product specific knowledge and PDP methods is set up
3.2 Established communication rules and time tables	4.2 Project manager and team members master CE methods
3.3 Infrastructure for communication with external team members (video conferencing, data exchange)	4.3 Decision making procedure is defined
3.4 Established traceability of communication history (accessible records)	4.4 PD team has supporting structure (prototyping, testing)
3.5 Proactive communication competences of internal and external team members	4.5 External teams are well integrated into core PD team
5 Process definition	**6 Organisation**
5.1 Defined product development phases with inputs and outputs	6.1 Project oriented organisation of product development
5.2 Systematic review of product development at milestones	6.2 Advanced level of project management (trained personnel, planning, supporting methods)
5.3 PD process definition is well understood and practiced by project teams	6.3 New product development team can be focused to project (splitting from existing production)
5.4 CE methods are built in the PD process and defined at level of details	6.4 Organisation enables fast and efficient cooperation with external teams and early involvement of suppliers
5.5 Project definition is integrated into project management and information support	6.5 Organisation is supporting smooth transfer of PD results into production

7 Information systems
7.1 Established product data management (PDM) for actual and past projects
7.2 Computer supported work and outputs at all phases (high level of visualisation enable communication)
7.3 Product development processes are integrated into PLM workflow configuration
7.4 Integration of external teams into information system
7.5 Integration of product development processes and data with other business processes

Fig. 22.4 Summary of the 7 assessment criteria

22.3.1 Interaction with Customers

As each product needs to fulfil customers' requests, permanent involvement of customers into PDP is essential. The whole cross-functional PD team needs to understand customers' requests. It is recommended that the whole core team participates in market research from the beginning. For end-user products it is necessary to involve users from specific markets in product assessment at PD milestones [2]. Kevin and Waleed [15] has proposed computer-aided product design that enables presentation of solutions and product details in a meaningful and understandable way to customers already in the conceptual phase. Assessment criteria for interaction with customers are summarised in Fig. 22.4. The first level is a written specification of customer's requests. Customers' involvement can be planned for product validation or milestones checking. The highest level is permanent customers' involvement during a PDP.

22.3.2 Involvement of Suppliers

Early involvement of suppliers into PDP is one of the key elements of CE if they are selected as long-term strategic partners. Innovative solutions in the conceptual phase of PDP have bigger potential than the cheapest supplier who is selected in the last moment.

A web-based PLM system can improve communication between customers and suppliers [16]. Volkswagen, e.g., has developed a methodology for efficient handling of requirements for mechanical parts [17]. Maturity levels for involvement of suppliers are presented in Fig. 22.4. At the first level suppliers are monitored according to the APQP methodology. This means that suppliers need to prove to customers at project milestones that they work according to the plan and to quality targets. Established information connection for document exchange can additionally improve communication between suppliers and customers. The highest level is active suppliers' participation throughout all PDP phases. The presented maturity levels can be treated independently from others. Long-term development of suppliers is not a guaranty or a request for implementation of the first four levels.

22.3.3 Communication

To support development, it is important to identify relevant communication channels as well as the frequencies and contents of communication [18]. The predominant type of communication varies considerably with the design level. At the new PD level, the world outside a company serves as an important source of information, while creative dialogue predominates. At the level of variants, designers are considerably more limited and dependent on the information that has been gathered and stored within the company. The distance between team members influences communication. The location of a development team in a common office or design studio stimulates informal communication (Fig. 22.5).

The following forms of communication have been recognized: creative dialogue, review and approval, informing team members and searching for information [19]. The type of communication varies with the phase of PD. During the conceptual phase many considerations are to be taken into account and harmonized, and many decisions to be made. This part of the process can hardly be formalized. Informal communication is very important, since it is the source of creativity. Physical proximity between project team members is the best way to accomplish a creative

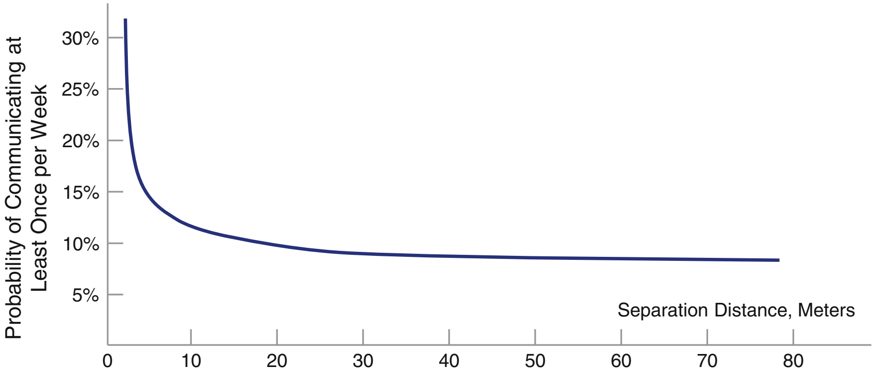


Fig. 22.5 Probability of communication and separation distance [2]

dialogue [2]. In later phase the range of people requiring access to product data becomes wider. Thus, access to the electronic forms of documents and communication via an information system is very important. Helpful are also web-based applications for managing informal communication based on 3D-mediated communication [20]. The results of communication analyses were used to optimize the processes and set the organizational structure. Assessment criteria for communication are summarised in Fig. 22.4. It is an advantage if a core team is collocated in one office. Communication with external or dislocated team members can be improved with equipment for video conferencing and data exchange. At the highest level, technical equipment is supplemented with communication competences of team members.

22.3.4 Team Formation

Before the start of a project preparation activities need to be executed as prerequisite of successful work. The goals should be set clearly; adequately trained individuals should be selected with care. For good co-operation, the team members should have complementary, and partially also the same, knowledge [21, 31]. Team members need to have specific skills, such as QFD, APQP, FMEA, SPC, design of experiments, six-sigma or other quality methods. The use of different methods stimulates cooperation inside a PD team for achieving better products.

The project leader has to adjust his role to the team's lifecycle phase [22]. The leader must ensure the clarity of goals, unanimous adoption of the work method and building of trust in the initial phases of the project, as well as encourage communication [23]. Each team member must be independent and must show initiative. Individual skills, such as, for example, knowledge of a foreign language in a multilingual team, cannot be mastered overnight, which should be taken into account as early as team formation [19]. Members of a team should together cover the paradigm of the product to be developed. It is the spirit, which introduces the product on the market. The paradigm is presented as a cloud of axioms. An interesting approach for CE task-team assignments is the use of cell-formation models and group-technology principles [24]. Maturity levels for team formation are presented in Fig. 22.4. The first level is the setting up of a well-structured multi-disciplinary PD core team. The second level requires that project manager and team members master CE methods besides technical knowledge. A clear and well understood decision procedure positively impacts team efficiency. At the fourth level importance has been recognized of the core team supporting structure with prototyping workshops and testing laboratory. At the highest level there is a smooth integration of external teams with the core PD team. It includes a formal procedure to activate a new external team and openness of the core team to new external ideas.

22.3.5 Process Definition

Implementation of CE requires a detailed process definition, which should be well understood by all participants. A PDP has characteristic milestones. Its execution should be in line with the company's special features and goals [2]. Examples of typical processes depending on the type of manufacture are presented in the next section. A common mistake in practice is to use the same process for small adaptive projects as well as for new PDs. This causes a great deal of waiting and long lead times during small projects—runners [25]. A clear division of processes and people who are in charge of them has proved to be successful. Workflow in an information system significantly contributes to traceability and transfer rates between individual workplaces [6]. It should be taken into account that development always involves a large degree of unpredictability. It also often turns out that additional research is necessary, as well as cooperation with external suppliers, customer approvals, etc. An effective PDP ensures reliable operation, especially in exceptional cases. Decision-making is often the bottleneck during a PDP. A good process also contains clear delimitations of competencies concerning decision-making and interventions in the case of complications. A multi-stage decision-making model is helpful in the case of complex products [26].

A PDP can include a specific methodology like a set-based CE approach [27], where a PDP starts with multiple design alternatives. Opposed to traditional design, set-based CE allows more than one design to proceed concurrently. The Design Structure Matrix facilitates a more structured approach at highly complex systems design [28]. Computer simulations and modelling of overlapping activities improve understanding of potential risks in complex processes [29]. Maturity levels of process definition are summarised in Fig. 22.4. The first level is the formal definition of a PDP. At the second level there is a systematic review at milestones of a PDP and triggering of needed corrective actions. At the highest level there is an integration of CE methods into a PDP and an integration of PDP with project management and information support.

22.3.6 Organization

The organizational structure should support CE during a PDP. It is important to set up a project organization that enables smooth execution of activities throughout all PD phases [6]. A functional or departmental organization provides deep knowledge but it can create also boundaries between different phases. An organization has to stimulate sharing of resources and tracking PD goals. For effective work, it is necessary to distinguish small projects on the adaptive design level from projects intended for developing new products [25].

Additional research can be time consuming. One of the possible solutions is setting-up flexible working hours, related to the amount of work. An additional useful measure is the sharing of employees (sharing resources) between departments or projects. This would mainly involve specialists for specific areas, e.g., surface treatment, noise and vibration, especially in the case of technically demanding products. It is recommended that team members should be prepared in advance to tackle typical problems. For good use of the capacities, it is necessary to ensure a good overview of the occupancy and flexibility when work is assigned. Each innovative organization should care for a network of external specialist with deep knowledge. It is important to be able to activate external experts fast without losing time with formal agreements.

Assessment criteria for organisation are summarised in Fig. 22.4. The first level is to set up a project-oriented organisation. An advanced level of project management implementation is a request for the second level. For the highest level it is necessary to have efficient cooperation with external teams and suppliers. A project organisation also needs to support smooth transfer of a developed product into production. There should be competent staff to take over the new product into production and experts from the core team need to be at least part-time available in the transitional period after transfer to production. Reference solutions are given in the next section for different types of manufacture.

22.3.7 Information System

An information system constitutes the necessary infrastructure for effective CE because during any PD PLM software has proved to be very suitable to support CE PD inside a company and also in a supply chain. Electronic communication is very convenient for informing team members, but it cannot replace a creative dialogue within the team [19, 31]. Configuration of a generic PLM system can be upgraded with a specific application like product relationships management for several life-cycle phases [30].

Documentation of an entire product family and planned variants are possible results of PD that need to be stored in and accessible with an information system. An extended product structure serves as the starting point for the preparation of variants at the variant and adaptive level of design [6]. The process for the preparation of variants can be considerably shortened when data and knowledge was documented already during PD. One must also make sure that new findings, e.g., from tests performed on the models, are also entered in the information system, so that they would be easily accessible later. The preparation of variants impacts the production process. Communication between variant developers and production people heavily relies on data that have been formally entered in the information system.

Maturity levels of information system are presented in Fig. 22.4. At the first level a product data management (PDM) system needs to be introduced. A PDM/PLM system is a meeting point for project team members where all project-related documents are available. At the second level an advanced level of visualisations tools is requested. Clear presentation of components and assemblies with 3D models and viewers is necessary for inter-team communication, especially in a spatially distributed environment. At the advanced level of PLM implementation PDP specific workflows are available for change management and for milestones approval. This level also involves external team members. At the top level the PDP is integrated with other business processes.

In the next section some examples of CE PDP are discussed, which have been adapted to individual types of production. These examples are based on the authors' many years of experience with the implementation of PLM and process analyses, as well as their personal experience from working in various companies.

22.4 Reference Models of CE Methods for PDP

Application of CE in PD needs to consider specific requests of different types of production. In machinery three types of production have been recognised.

- Individual production (CE—DIP)
- Serial production of modules or elements (CE—DPME)
- Manufacture of mass products (CE—DMMP)

Application of CE in PD is different for individual and serial production. Individual production is presented with an example from tool manufacturing. There are additional specifics in individual production of steel constructions or machines. The case studies demonstrate important characteristics and a systematic approach that helps to recognise details important for CE. The approach is sufficiently generic to be applicable to different companies.

There are more similarities between serial production of modules and manufacture of mass products. There is an example for serial production of vacuum cleaner motors. A typical customer is a company that integrates the modules into final products. Manufacture of mass products is typically done for end users. Therefore, attention needs to be put to marketing and understanding of end-user needs. An example is presented from household appliances.

Reference models have been set up on the basis of analyses of several real companies. Analyses were done during application of CE methods and PLM software by the authors. Good practice was upgraded with new technology and methods. Reference models run through several iterations. They were used at CE and PLM projects and then they were improved at each next project.

The reference models for CE methods include matrices with recognised specific questions and recommended tools and methods in each PD phase (Figs. 22.8, 22.11 and 22.13). General PDP is presented with characteristic details for individual and serial production. The reference models interconnect PDP phases and recognised CE methods as presented in Sect. 22.3. Each CE method or CE criterion has five maturity levels. In the reference models it is specified what level of maturity of a CE method has to be implemented at each PDP phase. The reference models need to be used together with case studies because they demonstrate in an illustrative way what is important and what different methods of CE have to be applied to each type of production. Application of a CE methodology has to start with analysis of an existing process. Added examples demonstrate how to recognise weak points and show ways for improvements.

22.4.1 PDP in Individual Production (CE—DIP)

Tool production is a typical example of individual production classified as variation or adaptive design. Design and production of tools require specific knowledge and equipment. Tools are recognised as products of minor complexity from the point of view of design, technology and number of variants point of view (Fig. 22.2). Tooling is an indispensable part of each serial production. It has, therefore, special importance.

In individual production (e.g., tool manufacture, machines, production lines and supporting structures) results have lower interdependency. Often, only the design documentation is prepared, while experienced workers themselves define the details of the manufacturing process. A decision on a change can therefore be a simple agreement between the chief designer and a production worker. However, direct decision-making during PD may be dangerous if it does not include an entry of changes in the design documentation.

All tools are produced for a specific customer, i.e., made to order. Production to order differs strongly from production to stock. Tool manufacturers are specialized in the technology used in the tool. The dynamics of tool technology development do not play a significant role, while this is on the level of variation design and, later, variation production. Components change only with respect to their dimensions, both in supplies and production, but their function does not change. A high degree of standardization and unified standards in production enables mutual cooperation between individual tool-making factories and the suppliers of materials and components. A sequential production method is characteristic for simpler tools, while work within larger projects and with shorter deadlines is performed concurrently.

In individual production, the boundaries between individual phases are blurred: the design engineer frequently participates in preparing tenders, orders the materials in advance, if necessary, and comes to the rescue if complications appear during production. The nature of developmental and design work has its own rules. The course of the work cannot be planned in detail. New perceptions are established during implementation, which result in certain changes. In individual production, unpredictability applies not only to design, but also to machining.

A general description of a tool production using a functional model (IDEF0) is presented in Fig. 22.6. The information flow or sequence of functions follows a product lifecycle from its ordering, designing, and production to shipping. Here the flow of events is followed irrespective of departments or individual workplaces. Each function or activity is represented by a rectangle designated by a name and its level in the hierarchy. Data or materials input takes place on the left side and output on the right. Mechanisms or resources are listed below the rectangle or function, and above the function are limitations and controls.

22.4.1.1 Analysis of Information Flow in a Toolmakers Factory

A critical assessment of the contents and review of paths have been performed for all documents, namely, in what sequence and where the contents have been created, confirmed and used. The interdependence of activities was analysed using cross-comparisons. Analysis of the production process shows that individual tasks must be performed in sequence [30]. For example, it is not sensible to begin preparing design documentation before formal confirmation of the contract. Preparation of a tender consists of the setting up requirements, a technical solution preparation, evaluation of the technological feasibility and calculations. The first step is the specification of requirements, after which the following steps can be performed concurrently. It is necessary to take into account, as early as the conceptual design phase, the costs and possibilities of production with respect to the available machine pool. On the other hand, the preparation of a tender for the tool is predictable and not time-consuming, making a sequential course of tender preparation often acceptable.

The first part of preparation of design documentation takes place in the conceptual design phase. At this point, the designer checks the tool as a whole and the interdependence between components using a 3D presentation. An extended conceptual design phase ensures that the tool structure remains unchanged later in the process, as does the independence of individual components. This enables concurrent work on individual parts of the tool and division of work between several designers. At the same time, one can already determine the size of preforms for critical parts and give the go-ahead for their ordering [25].

Analysis of the work process shows the phases that may be overlapped: tender preparation and design, production of parts and assembly, while the purchasing department for parts and components must have a good response time. The conclusions of analysis are listed below.

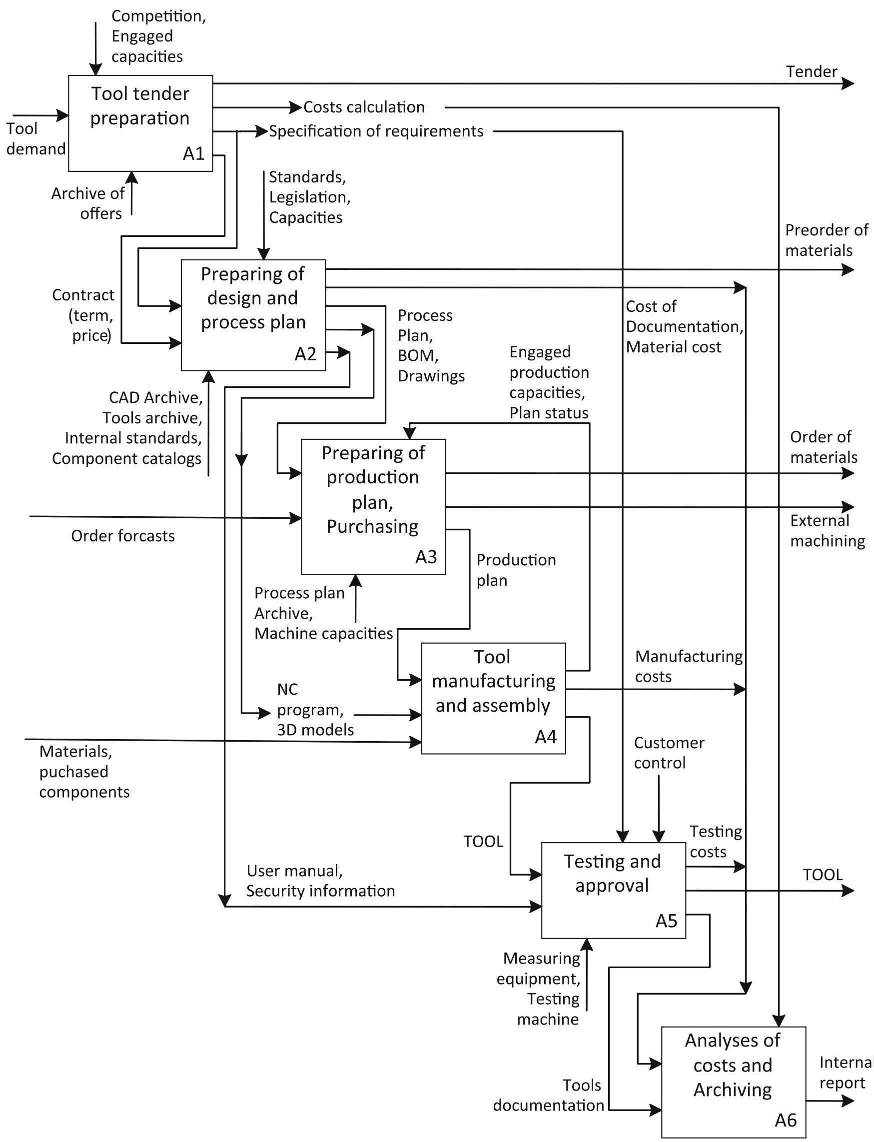


Fig. 22.6 Functional analysis of tool production (IDEF0)

- Process chain through the entire tool production process from order to delivery.
- Close link between the development of serial products and tool production.
- Information connection and close relationship with key component and material suppliers.
- Products for individual production should be built in a modular way, with a reliably working core, which makes the adjustments more predictable.

- The information system guarantees a good overview of the situation in a company—delivery times, current situation in the production process, recorded mutual exchange of information.
- It is necessary to ensure that individual decisions in the development process are recorded, so that the general picture is not in an individual's mind only.

Some of the recognised potentials for improvement of PDP in individual production were found as good practices in successful companies. The other improvement options are conclusions from information flow analyses and assessment of emerging information technologies and methodologies. The improvement potentials presented bellow in a comprehensive form is a desired situation for individual production. Each company needs to analyse its PDP individually and set the importance and optimal implementation way of suggested improvements.

22.4.1.2 Process Chain Through the Entire Tool Production Process from Order to Delivery

As there is no sharp boundary between the individual phases from tender preparation to manufacturing, the possibility must exist to track individual parts or assemblies throughout all phases of production, independently of each other. In the conceptual design phase, tools are broken down into autonomous and independent parts and assemblies as much as possible; these are then dealt with separately. This enables concurrent performance of several activities. Information systems make reviewable tracking possible of each individual building block, so that separate work on building blocks does not take extra time or reduce reviewability. Function and functionality, and their geometrical inclusion into space, are determined for the parts and assemblies so defined. For each individual part, the phases of detailing, determination of technology, purchasing and production take place in succession. For the tool as a whole, overlapping occurs, which reduces total production time.

22.4.1.3 The Connection Between the Development of Serial Products and Tool Production

In serial PD, it is important to prepare a detailed plan of technology, which in serial products is basically connected with tool production. The development and testing of products for serial production is, as a rule, a time-consuming process. However, the majority of products change very little during the final phases of PDP. This means that the design and production of individual tool parts can already begin considerably before final serial product confirmation. The total tool production time will not be shorter, but regular tool production can begin much earlier. The majority of tool components can be ordered and produced in advance. Only those parts that give the tool its final shape wait for serial product confirmation. More total available time makes possible a cheaper supply of materials and their more uniform distribution on machines.

The overlapping of serial PD and tool production requires greater attention. If a toolmaker factory is an integral part of a company developing a serial product, the ensuring of data exchange and a concurrent work process is not so problematic. In the case of external contractors, it is necessary to supplement the established working method, whereupon all details and the price will have been defined by the time the contract is signed. In addition to a constant exchange of actual information, a high degree of mutual trust is necessary; this guarantees quality and supply at market prices to customers, while at the same time, the toolmaker factory is guaranteed preliminary information on orders.

22.4.1.4 Information Connection with Component and Material Suppliers

With tools, the deadline for the delivery of materials has an important influence on total production time. The use of materials and components needs to be standardized inside the company to the greatest extent possible. The setting up of a direct connection with suppliers both reduces the delivery deadline and automates ordering procedures. In this way, suppliers are already receiving data on planned materials during the period of conceptual design of the tool, so that they can adapt their stocks to future needs and reduce delivery deadlines to toolmakers factories. The supplied materials include data on quality, price, and date and method of delivery. Information on the purchased materials can thus be included directly into information support systems. The costs of materials represent only about 10 % of a tool's price; therefore, a time-consuming collection of tenders and ordering of materials and components are not justified. Suppliers of internally standardized materials and components are included directly in the information chain. The most expensive parts are ordered by collecting tenders.

The data model in Fig. 22.7 shows that an object Process plan that unites the required machining operations on various machines is bound to each structure element. The data model for a structure element consists of data that remain unchanged during repetition (workshop drawing, technology) and of operational data on purchasing and monitoring of production. Currently relevant operational data are bound to the structure element with a relation Production, as distinct from their relation Finished, which already contains completed machining.

22.4.1.5 Reference Model for CE Methods for Individual Production of Tools

As indicated above, the tool development process needs to distinguish complex tools from typical tools. For a complex tool it is critical to select the right concept. A systematic approach with team concept assessment and use of numerical

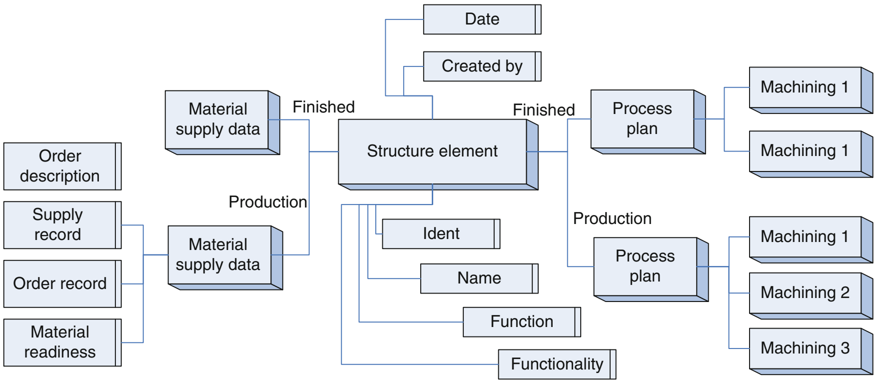


Fig. 22.7 Data model for structure element and monitoring of production

simulation of injection moulding or stamping helps to take the right decisions in the Conceptual phase and on System level design (Fig. 22.8).

For simple tools a short manufacturing time is needed against competitive costs. In the first phase support is needed for fast and reliable preparation of proposals (offers) to the customer. In the reference model in Fig. 22.8 there is the request for Support for tender preparation at the Planning phase. The whole procedure from tool design to manufacture needs to be smooth and well supported with a minimum number of iterations.

In Fig. 22.8 CE methods are linked with PD phases. During the planning phase communication between the tool designer and the serial product developer is critical (e.g., small modification to a serial product can make an injection moulding tool simpler and the process more reliable). In the reference model in Fig. 22.8 minimal maturity level requests are set for the assessment criteria from Fig. 22.4. In the Planning phase a written specification is requested for a new tool (1.1) and direct contact between toolmaker and customer (1.2). Open communication is possible if the toolmaker is selected as long-term strategic partner (2.2). It is important that the toolmaker core team put requests on the table and discuss them (3.1). Easy access to earlier tool projects is very helpful (7.1). After the conceptual phase tool architecture assessment by the lead designer needs to be assured to avoid faults and modifications in the next phases. This request is given in the reference model with Team formation criteria 4.1 and 4.3. Tool design for most of the tool orders is executed inside the golden loop design process (Fig. 22.3). A tool request from a customer enters the process and it need to be transformed into a tool specification. From time to time the toolshop takes orders that require running the invention and research loop. A trigger for a research loop can be also a management decision. A toolshop wants to make a step forward in tool architecture, machining technologies, internal logistics or in computer-aided support. Inside the innovation and research loop the toolshop collects new knowledge, while new methods are

Product development phase	Specific requests for CE methods	Requested level of CE criteria	Tools and methods
Planning phase	Customer involvement: Communication between tool maker and product developer (customer)	1.1, 2.2, 3.1, 5.1, 7.1	Support for tender preparation, APQP
Conceptual phase	Communication: Reuse of knowledge from realized projects (tools).	1.2,1.5, 3.1, 4.1,4.3, 5.1, 7.1,7.2	Numerical simulation
System level design	Team formation: Senior tool designer need to participate in tool architecture assessment	3.1, 4.1, 5.2, 7.1, 7.2	Design-FMEA Team assessment of tool design concept
Detail Design	Information system: Use of standard components	4.1, 4.2, 6.3, 7.1, 7.2	PLM, library of standard elements
Testing and validation	Customer involvement: Execute tests with customer in toolshop	1.3, 5.1, 5.2	Statistical control, Measurement system analyses
Production process planning	Information system: Simple and efficient structure element tracking through production	6.4, 6.5, 7.1, 7.2, 7.4	
Production launch, distribution and service	Process definition: Working instruction, Spare part need to be well documented	7.1, 7.2, 7.4	
Product disposal at the end of product life cycle	Process definition: Environment awareness	4.1	

Fig. 22.8 Reference model for CE methods for individual production of tools (CE—DIP). Numbers in column "Requested level of CE criteria" are taken from Sect. 22.3 from assessment criteria. Example: "1.1" means "Written specification of customer's requests"

integrated into a new product (tool) development process. This step can be comparable with a new product family development process at serial production as presented in the next two sections.

22.4.2 PDP in Serial Production of Modules or Elements (CE—DSPME)

Modules or elements are generally not sold to end users but to customers that integrate modules into final products. Customers therefore speak a technical language. It makes communication between supplier and customer/integrator easier. Anyway, a manufacturer of modules cannot ignore end-user needs. They can help understand technical request better. It is recommended to include the end users into the development process of modules. It is important to have technology and products' characteristics on a high level; industrial design is for modules and elements, which cannot be seen by end user to be of secondary importance. As there is always a cost pressure, product design and manufacturing processes need to be optimized. Customers/integrators of final products expect support for the introduction of innovations and new features of modules. Integrators want to focus their R&D resources to final products. Trends and new features of modules have to be set by the module manufacturer. There is a case study below on a vacuum cleaner motor, which is a key module for vacuum cleaners.

Case study from the Domel company: vacuum cleaner motor.

Vacuum cleaner motors are an example of variation design. Electric drives have a medium level of complexity from the design and technology point of view (Fig. 22.2). In product planning a new module development causes a big investment in tooling and the assembly line. It can be profitable only if a critical production quantity is guaranteed for a specific period. There are several key customers who need to be interested in most of the new products. There should be a distinction between new product generation development and customer-specific variant design.

A new product generation is kept in production from 4 to 7 years and it is an example of innovative design. Adaption or variation of an existing module or element in serial production to specific customer requests is happening inside the golden loop design process (Fig. 22.3). A new product generation development for serial production of modules or elements is presented in Fig. 22.9.

The core team for a new vacuum cleaner motor is usually small (ca. 3 members). Internal communication, therefore, is not problematic. It is not realistic to collocate the extended team (tooling, assembly line, process planning). Extended team members are working on the project part-time only. Specific knowledge and methods for the vacuum-cleaner development are presented in Fig. 22.10.

New PD is divided into a research and development phase and a design phase (Fig. 22.9). The role of the research loop is to set up technical specifications for the new product on the basis of systematic research. Research may include functional prototypes, deep research on new materials, key components and new assembly principles. A feasibility study requests specific knowledge.

It is a challenge to integrate deep technical knowledge (aerodynamics, noise and vibration, electric design, automation, diagnostic methods) from external research groups into a new generation of products. There may be several external team members with specific knowledge that belong to different functional units (departments). Technical specifications, which are defined at the planning phase and updated during the research phase, are entering the golden loop PDP.

At each PDP phase specific knowledge and recognised working methods need to be used. In Fig. 22.9 on the left side specific knowledge is presented and on the right side working methods. There are also several iterations at each development phase. The focus is first of all on details related with CE.

Mapping between CE methods and module or element development phases is presented in Fig. 22.11. The reference model for CE methods for production of modules and elements (Fig. 22.11) contains a required minimum maturity level of

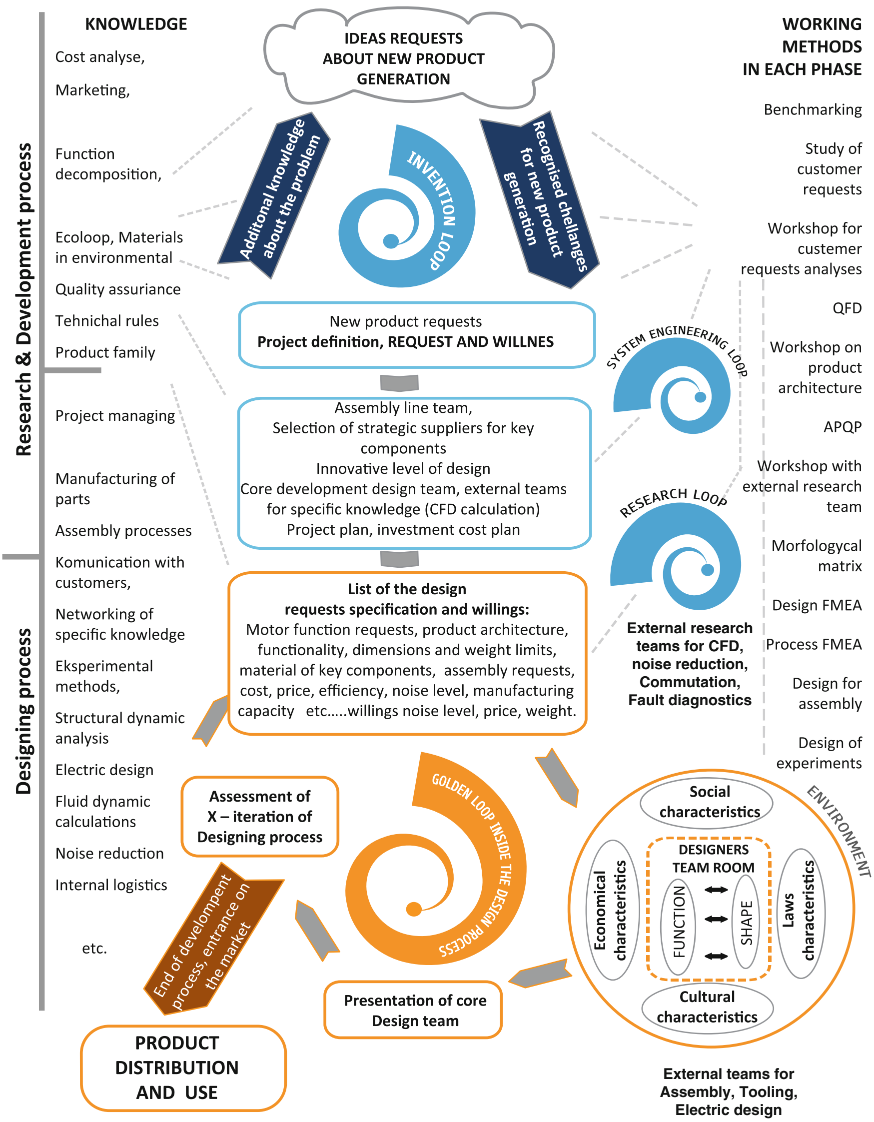


Fig. 22.9 Case study: PDP of electric motor with specific details

CE criteria. Look for example at detail design of PDP. There is a need for close cooperation with suppliers of key components (2.1, 2.2, 2.3 and 2.5). For professional execution of detail design a high maturity level of communication is required. There is a request for all maturity levels of team formation and the first three levels of process definition (5.1, 5.2, 5.3). At process definition, organisation and information system a top maturity level gains in importance for bigger teams

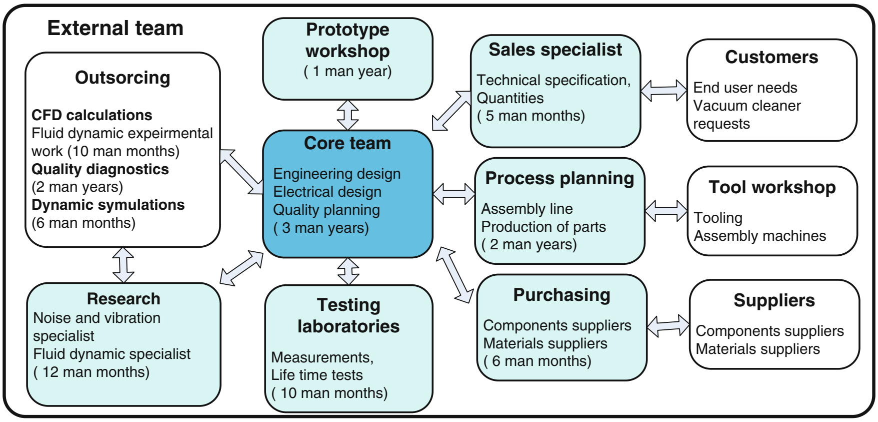


Fig. 22.10 Structure of core and external team for vacuum cleaner development

and product complexity. The core team needs to get support from external team members. Collection of specific knowledge can be critical. Active participation of external team members is provided through workshops. The QFD, FMEA workshops have predefined time schedules, procedures and reporting of results. If necessary, additional team or individual meetings can be planned. While another set of activities is preserved for external team members, they get all necessary additional information from core team members (Fig. 22.10). Advanced testing and measurements on prototypes are an important source of information. There should be no hold-ups in the serial production process. Since interaction between different fields is much higher in this case, the production process must be carefully planned and the communication channels must be provided for. It is vital to ensure coordination and cooperation between the development process, production arrangement (technology, tool manufacturer), production process and the company's management. Development of different technological and constructional details is continuous.

Modules include years of experience and special elements. In terms of design and development of an individual structure, an individual or a small development group can usually maintain the overall view of the entire process. Designers are specialized in individual types of products and they are very well acquainted with constituent elements and technology of their design. Knowledge management plays an important role. The company should be able to transfer knowledge from one product generation to another.

Adaptation of a product to specific customer needs is an iterative process at the adaptive or variation design level. The change committee, together with members from technology, quality, production, supply, sales and service departments, should examine proposals from all perspectives and either approve or reject them. In cases of ambiguity, further studies are required. The change committee should set a date when a change will take effect; subsequently the proposal will be implemented.

Product development phase	Specific requests for CE methods	Requested level of CE criteria	Tools and methods
Planning phase	Customer involvement: Understanding of customers' needs	1.1, 2.1,2.2, 3.1,3.2, 4.1, 5.1, 7.1,7.2	QFD, APQP, Benchmarking Technical specification
Conceptual phase	Organisation: Collection of specific knowledge from external sources and teams (noise reduction, electric design, fluid dynamic)	1.2, 1.5, 2.4, 3.1, 3.2, 3.3, 3.4, 4.1, 4.2, 4.3, 4.5, 5.1, 7.1, 7.2	Specific research, numerical simulations (CFD, structural dynamics)
System level design	Process definition: Team formation with external team members; deep know-ledge on product technology, tooling, assembly, diagnostic is needed.	1.1, 1.5, 2.5, 3.1, 3.4, 4.1, 4.2, 4.3, 5.1, 5.2, 7.1, 7.2	Moderation of workshops Six sigma methods
Detail Design	Organization need to support creation and work of cross-functional teams. Involvement of key suppliers into PDP	2.1, 2.2, 2.3, 2.5, 3.1, 3.2, 3.3, 3.4, 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 6.1, 6.2, 6.3, 6.4, 7.1, 7.2, 7.3	Design-FMEA Design of Experiments
Testing and validation	Communication Testing result need to be well understand in the team	1.3, 4.1, 4.4 5.1, 5.2	Measurements (dimension, electric parameters, efficiency, noise level) testing, life span testing
Production process planning	Information system: Simple and efficient structure element tracking through production	4.1, 4.3, 6.5, 7.1, 7.2, 7.3	Process_FMEA, Quality planning
Production launch, distribution and service	Processes definition should enable smooth end efficient work.	7.1, 7.2, 7.5	SPC, MSA, Product diagnostic
Product disposal at the end of product life cycle	Process definition: Material selection, disassembly procedures need to be considered in conceptual phase.	4.1, 4.5	

Fig. 22.11 Reference model for CE methods at serial production of modules and elements (CE—DSPME). Numbers in column "Requested level of CE criteria" are from Sect. 22.3

A good communication with external component suppliers and tool manufacturers is especially important to ensure short delivery times. The virtual 3D model is the most suitable for remote communications.

The question of a demarcation line between electronic and direct (personal) communication is an important one. According to the analysis [6, 19, 30] creative dialogue dominates in invention, system engineering and research loops, while an information system (PLM, ERP) functions best for its implementation at the adaptive design level. Each adaptation to customers needs to be treated with special care because a product family in serial production is sensitive to modifications [6]. Technical changes should be approved in a working meeting, since a creative dialogue brings about the synergy effect and the work is usually faster. A PD team needs to plan a possible product adaptation already in the system-engineering loop. Product structure elements have to enable modular exchange and further product evolution processes (see Chap. 14). Product results do not have only technical characteristic. A product family will be manufactured in a period from 4 to 7 years with several hundreds of variants created for different customers. Therefore a product structure and the structure of technical documentation have to be well documented in the technical information system (PLM). Product configuration management needs to be integrated in the planning phase of the PDP. Implementation is part of the golden loop design process.

Although a vacuum cleaner motor is of moderate complexity concerning the number of variants and components, its PDP has all characteristics of a PDP for complex systems. The phases of decision-making and document distribution are critical. An orderly information system is a large advantage both in terms of costs and reliability. The majority of changes require professional analysis, production of prototypes and measurement of technical characteristics. The company's organizational structure should enable quick responses to customer requests.

22.4.3 PDP in the Manufacture of Mass Products (CE—DMMP)

Case study for cooler development (Gorenje)

A PDP of household appliances is in general similar to a PDP of serial production of modules (Fig. 22.9). Important differences are presented in Fig. 22.12 and specific requests for CE methods at different PD phases are shown in Fig. 22.13. As household appliances are produced for end users, deep understanding of end-user needs in the whole PD team is of key importance. Several supporting activities are executed to collect needed information from end users and translate them into technical specification. The complexity of products and PD teams for household appliances are bigger, requiring a specific approach for enabling efficient communication. The production life span of a mass product is in general shorter compared to modules or elements. Industrial design and fast introduction to market are of dominant importance. Fast execution of a complex project is a challenge. Additional attention has to be put to team formation: core team members need product-specific knowledge and they have to be compatible also from a personal character point of view. The PDP described applies to a new generation of PD (innovative level of design). At the same time it is important to be

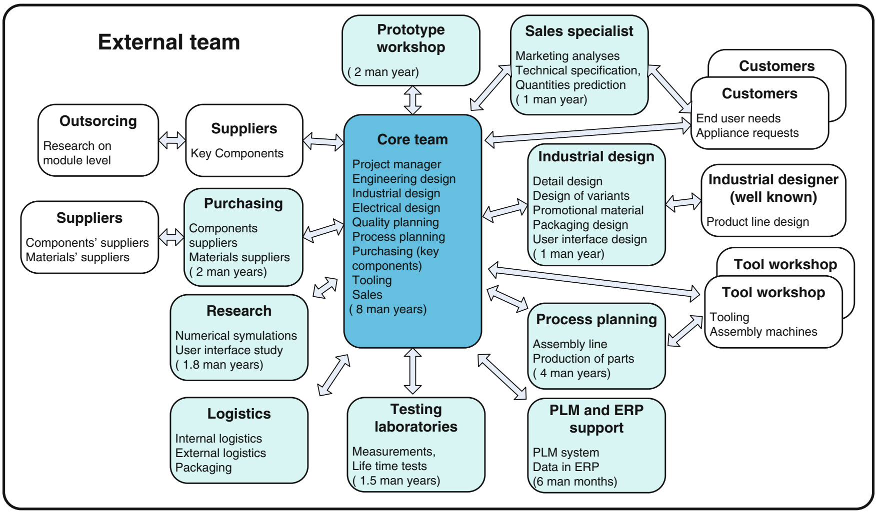


Fig. 22.12 Structure of core and external team for home appliance development

aware that the basic product family needs to be many times adapted in detail to different specific customer requests. The modular structure of a product and technical documentation has to enable fast and reliable management of variants derived from product family.

The expected time to execute home appliances development projects in man-year is longer than for modules or elements. Bigger core teams should be collocated in the same office to enable smooth inter-team communication. Some of the activities for vacuum cleaner motor development, which were managed before by external team members located in functional departments, are now managed by core team members. For example, industrial designers inside a core team communicate directly with end users, a specialist for process planning has direct contact with tool suppliers. Direct communication reduces the probability of wrong data interpretation and improves the transfer of information inside the core team. It is known that increased physical distance between employees' workplaces dramatically reduce the frequency of informal communication and data transfer [2].

Key competences of a PD team are integrated PD, marketing and logistics management. It is expected that deep knowledge of specific components is managed by key suppliers. On the one hand, inside the complex team it is necessary to preserve smooth communication through inclusion of multiple functions. On the other hand it is necessary to split small independent tasks and execute them in smaller teams. It would be simply inefficient to involve all external team members in every decision.

Each core team member is expected to have the whole picture of PD. The project manager has to provide information transfer to team members with periodic core team meetings twice a week or once a day during specific project phases like

Product development phase	Specific requests for CE methods	Requested level of CE criteria	Tools and methods
Planning phase	Interaction with customers: Understanding of end user needs by all core team members Process definition: well defined PDP Team formation: It is needed specific knowledge and compatibility on personal level	1.1, 1.2, 2.1, 2.2, 2.3, 3.1, 3.2, 3.4, 4.1, 5.1, 5.3, 6.1, 6.2, 7.1, 7.2	QFD, APQP, benchmarking, Interview / workshops with end-users
Conceptual phase	Organization: core team is collocated in the common office; well defined work with external teams. Early involvement of industrial design	1.2, 1.5, 2.1, 2.3, 2.4 3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.2, 4.3, 4.5, 5.1, 5.2, 6.1, 6.2, 6.3, 6.4, 7.1, 7.2	Industrial design, Systematic work with suppliers
System level design	Organization: attention on inter team communication: Communication with external team members is done through core team members	1.1, 1.5, 2.5, 3.1, 3.4, 4.1,4.2,4.3, 5.1, 5.2, 6.1, 6.2, 6.3, 6.4, 7.1, 7.2	Industrial design, Six sigma methods
Detail Design	Organization need to support work in cross-functional teams. Splitting of working tasks. Involvement of suppliers: Urgent is early involvement of suppliers into product development process Information system: technical documentation is in structural way kept in the PLM database	1.4,1.5, 2.1, 2.2, 2.3, 2.5, 3.1, 3.2, 3.3, 3.4, 3.5 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 5.4, 5.5, 6.1, 6.2, 6.3, 6.4, 6.5, 7.1, 7.2, 7.3, 7.4	Design-FMEA, Workshops for to stimulate inter team communication
Testing and validation	Customer involvement: Beside technical characteristics it is very important to get feedback from end users	1.1,1.3,1.4, 2.5, 4.1, 4.4, 5.1, 5.2	Design of Experiments Product testing
Production process planning	Focus on assembly line and logistics	4.1, 4.2, 4.3, 6.4, 6.5, 7.1, 7.2, 7.3	Process FMEA
Production launch, distribution and service	Information support: Product data need to be managed well in production phase.	6.5, 7.1, 7.2, 7.5	MSA, SPC, PLM, Flexible workflow
Product disposal	Process definition: Material selection, disassembly procedures need to be considered in conceptual phase.	4.1, 4.5	

Fig. 22.13 Reference model for CE methods in the manufacture of mass products (CE—DMMP). Numbers in column "Requested level of CE criteria" are taken from Sect. 22.3

conceptual design. A core team member has to pass the necessary information to external team members so that they are able to work on a project with alignment to the whole project. All formal meetings and decision-making are recorded.

As external suppliers have a key impact on PD, it is necessary to involve them early in PD. Small technical details of components can influence product characteristics. Serial production of household appliances is usually based on an assembly of elements and modules produced by different suppliers. The quality and timely delivery by several suppliers should be guaranteed. Umbrella companies should be in charge of marketing and development of end products. Product-development time is reduced by the transfer of development information on a module and component level to strategic suppliers.

After production launch, adaptation of existing products to specific customers is a type of development process on the adaptive or variant design level. In the manufacture of household appliances the range of possible technical changes may become so huge that it is sensible to form groups for characteristic types of product adaptions, e.g., sheet metal, plastics and surface treatment. In this way the working process in smaller groups is more effective. Flexibility can be achieved in different ways: a group of selected specialists can be called according to the problem; a virtual group is defined throughout the flexible workflow. The documentation about changes should be transparently accessible in the information system.

In the manufacture of household appliances, there are many design-related changes (consumer needs). From a technical point of view, it is more difficult to control a vast number of changes and the entire logistics than individual changes. A change of documentation simultaneously also comprises a feasibility study in order to reduce PD time. Long lead times and poor communication in the chain are obvious. With PLM software and workflow configuration, the two-phase approach becomes established: change review and approval in the first phase, and entry of the change in the documentation in the second.

Based on the analysis of household appliance manufacturing [6], we have found that the approval regarding the feasibility of a change in a two-phase chain is not optimal. The most effective way for making a definite decision for change approval is a creative dialogue between the team members. The dialogue can be conducted by means of a videoconference when locations are physically remote. A flexible workflow is vital for the process of modifying documentation. There is a high degree of unpredictability in technical change management. Therefore, the workflow must be flexible, so that the way can be defined simultaneously, according to the needs. An overview of each individual document's status should be provided in terms of its current location. Easy access and user-friendliness are important. A product adaption should be implemented in a predefined sequence. However, those included in the process should be able to consult anybody, including external suppliers. In this way, a virtual group is formed and it can function effectively, as if it was located in the same place.

With the large number of variants and also of participants in the process, ICT support becomes indispensable for communication. No individual alone can have a good overview of the numerous processes that take place simultaneously. A PDP at the variation design level must be supported by a flexible workflow, so that each participant receives only those documents with which he or she needs to perform some activities. The inclusion of external suppliers in the information system is especially important for a good flow of information and effective decision-making, so that these can be independent of the location.

22.5 Discussion on CE Criteria and Reference Models

The authors cannot assure success from applying CE methods just by following the reference models. Change of a PDP is a complex task. The first prerequisite is to have a will to improve PDP at the top-management and operational level. The next step is to acquire competences on new methods. The personnel is expected to have confidence in new methods and planned changes when they have convincing knowledge of them. The supervision of personnel by an expert during the introduction of CE methods can be very helpful. The expert needs to have a vision and examples of successful CE implementations.

When the preconditions are met the CE project can start. The core of everything is a PDP analysis of the current situation. The presented case studies and reference models can be used to identify the gaps. The company has to recognize its strong and weak points. It is not possible to prepare a simple procedure for implementation. The reference models represent a mirror for a company to check and compare the current or already modified processes.

Implementation of CE methodology can have several different ways and focuses. There are enterprises that have achieved a push forward in PD by introducing advanced project management. Others can see an important contribution by applying six sigma methods. The selected seven CE criteria represent a balance between different CE methods and are in the core of each application of CE to PDP. Attention should be put to the maturity level and contents of each criterion. Project management is comprised in the organization block; use of methods is part of team formation and team members' competences.

An example of reference model usage for individual production is presented below. According to the data in Fig. 22.4 the requested CE criteria for the planning phase are listed below.

1.1 Written specification of customer's requests
1.2 PD team participation at collecting of customer's requests
2.2 Suppliers are selected as long-term strategic partners
3.1 PD core team is collocated in one office
5.1 Defined PD phases with inputs and outputs
7.1 Established PDM for actual and past projects

There is additional request for close contact between tool maker and product developer. Cooperation is much easier if the tool maker is selected as long-term strategic partner. Computer-aided support for preparing tenders makes this task faster and more accurate. A salesperson for tenders needs deep knowledge on tooling and he has to be collocated with tool-design specialists. Access to past projects can accelerate preparation of tenders and improves the overview of existing knowledge.

In individual production of tools there are specialists who have expertise on tooling. In serial production team formation and use of specific methods are much more important. In serial production additional attention has to be put to benchmarking and understanding of end-user needs. All these details can be inferred from the reference models. Use of a reference model is more accurate if it is used after PDP modelling of the current situation. It is recommended to perform an assessment of each criterion in the following way: 0—the criterion is not implemented at all, 1—there is a modest use of the specified criterion, 2—the criterion is practised in every day work, 3—the implementation of the criterion is at top level, it can be used as a reference for the others. The criteria in the reference models can be used for periodic assessment of the CE level during a continuous improvement process. The target values for each PD phase for all three reference models are presented in Fig. 22.14 as a percentage of the maximum possible maturity level. Target values are calculated from reference models in Figs. 22.8, 22.11 and 22.13. A 100 % would be accived in the case that all CE criteria are at top maturity level (assessment 3). An implementation of each requested CE criterion in the analysed company can be assessed from 0 to 3 points.

The list of CE criteria and maturity levels has been collected through several iterations. During testing it has been approved that they now cover needs for different cases of all three types of production. In the next years the reference models need to be updated with new methods and IT technology. It will also be a

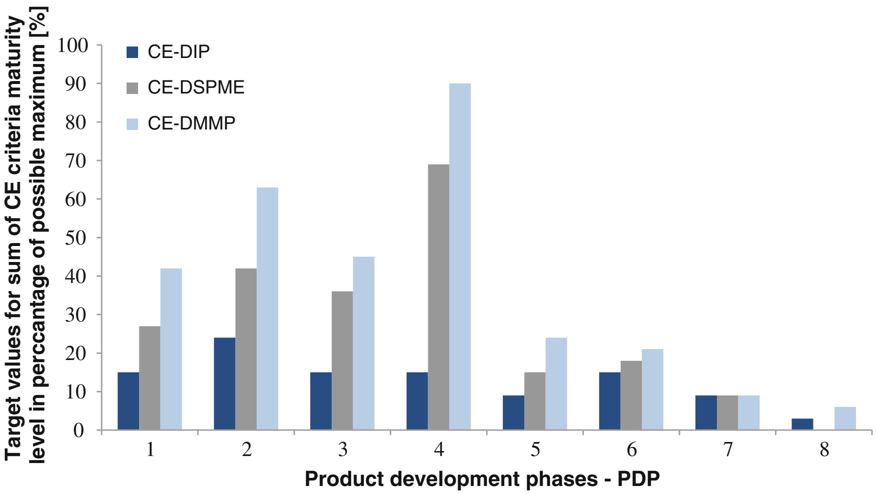


Fig. 22.14 Target values for sum of CE criteria for DIP—individual production of tools, DSPME—serial production of modules and elements and DMMP—manufacture of mass products in each PD phase: *1* Planning, *2* conceptual phase, *3* system level design, *4* detail design, *5* testing and validation, *6* production process planning, *7* production launch, *8* product disposal

continuous improvement process. The CE assessment model will be updated step by step on the basis of collected experiences of applying CE and with new tools and methods and imaging IT technologies.

22.6 Conclusions and Outlook

This chapter presents specific criteria that need to be fulfilled for CE to be managed well: involvement of customer and suppliers, process definition, communication, team formation, organization and information system. The impact of an individual criterion depends on the type of production, product complexity and PD phase. The specific requests of different types of production with its PDP phases are presented in the reference models for CE methods.

Reference models for CE methods for individual production, serial production of modules or elements and manufacture of mass products help to understand the application of CE. The reference models are the result of systematic research and personal experiences of the authors. Specific CE methods and criteria are presented first in general form. It has a limited practical value for engineers who are applying CE methods into a PDP. Because each company has many specific characteristics and constraints, it has to find its own way of implementing CE. Reference models have not been prepared as a prescriptive method to be applied step by step. They present case studies on systematic PDP analyses and what needs to be taken into consideration. This chapter contribution is deeper understanding of CE methods in relation to product complexity, design level and production type. The reference models for CE methods need to be used together with examples because they show what is important.

Additional contribution of this chapter for better understanding of PD in relation to CE is a general as well as a specific presentation of a PDP. A clear definition of the process and organization of a design process has importance for all PD phases. A generalized model of the PDP with a golden design loop helps to understand key phases, iteration loops, needed knowledge and methods. It is important to distinguish between research and design phase of the project. In the planning phase, which includes the system engineering and research phase, the new product idea should be transformed into a project definition. Product design is finally executed inside the golden loop. CE principles are included inside several iterations or loops. When a conceptual design is created inside the golden loop it is checked several times against all requests. If there is a finding at an assessment point that a product design is not yet satisfying market requirements and specifications the design iteration is repeated. In some cases the process goes back to research and the system engineering loop.

The application of CE methods to a PDP is a continuous process. Methods have to take advantage of new information technologies. The company profile and production program is changing continuously causing concurrent adaptation of the way CE methods are implemented.

References

1. Duhovnik J, Tavčar J, Koporec J (1993) Project management with quality assurance. Comput Aided Des 25(5):311–320
2. Ulrich KT, Eppinger SD (2012) Product design and development. McGraw-Hill, Boston
3. Duhovnik J, Starbek M, Prasad B (2001) Development of new product in small companies, Concurrent engineering: research and applications, vol 9. Technomic Publishing Company, Inc., Lancaster, pp 191–210
4. Zadnik Ž, Starbek M, Duhovnik J (2012) Enhancing preliminary design within concurrent engineering using the matrix of functions and functionalities. Concurr Eng Res Appl 20 (4):275–285
5. Benedičič J, Žavbi R, Duhovnik J (2012) Development of a new method of searching a new product development opportunity. Teh vjesn 19(4):759–767. http://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=137676
6. Tavčar J, Duhovnik J (2005) Engineering change management in individual and mass production. Robot Comput Integr Manuf 21(3):205–215. http://www.sciencedirect.com/science/journal/07365845
7. Zadnik Ž, Karakašić M, Kljajin M, Duhovnik J (2009) Function and functionality in the conceptual design process. Stroj Vestn 55(7/8):455–471
8. Duhovnik J, Žargi U, Kušar J, Starbek M (2009) Project-driven concurrent product development. Concurr Eng Res Appl 17(3):225–236
9. Duhovnik J, Kušar J, Tomaževič R, Starbek M (2006) Development process with regard to customer requirements. Concurr Eng Res Appl 14(1):67–82
10. Prasad B (1996) Concurrent engineering fundamentals: integrated product and process organization, vol I. Technomic, Lancaster, NJ
11. Lawson M, Karandikart H M (1994) A survey of concurrent engineering. Concurr Eng Res Appl 2(1):1–6
12. Ainscough M, Neailey K, Tennant C (2003) Charles A self-assessment tool for implementing concurrent engineering through change management. Int J Proj Manag 21(6):425–431
13. Hrzek V, Macke N, Luekens E (2013) Qualitative assessment of business processes, In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on con-current engineering. Springer, London, pp 883–894
14. Shouke C, Zhuobin W, Jie L (2010) Comprehensive evaluation for construction performance in concurrent engineering environment. Int J Proj Manag 28(7):708–718
15. Kevin TC, Waleed EG (2012) An intelligent system based on concurrent engineering for innovative product design at the conceptual design stage. Int J Adv Manuf Technol 63(5–8):421–447
16. Tavčar J, Potočnik, U, Duhovnik J (2013) PLM used as a backbone for concurrent engineering in supply chain. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 681–692
17. Guemmer G, Junk C, Rock G (2013) A variant management based methodology for the requirements-engineering process of mechanical parts. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 109–120
18. Žavbi R, Tavčar J, Verlinden J (2008) Educating future product developers in virtual collaboration: five years of the E-GPR course. In: Kisielnicki J (ed) Virtual Technologies: concepts, methodologies, tools, and applications. Information Science Reference, Hershey PA, pp 101–128

19. Tavčar J, Žavbi R, Verlinden J, Duhovnik J (2005) Skills for effective communication and work in global product development teams. J Eng Des 16(6):557–576. http://www.tandf.co.uk/journals
20. Siltanen P, Valli S (2013) WEB-base 3D mediated communication in manufacturing industry. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 1181–1192
21. Rihar L, Kušar J, Duhovnik J, Starbek M (2010) Teamwork as a precondition for simultaneous product realization. Concurr Eng Res Appl 18(4):261–273. doi:10.1177/1063293X10389789.22
22. Pirola AM, Härtel C, Mann L, Hirst G (2002) How leaders influence the impact of affective events on team climate and performance in R&D teams. Leadersh Quart 13(5):561–581
23. Esra A, Nallan CS, Li L (2011) Team formation in concurrent engineering using group technology (GT) concepts. Concurr Eng Res Appl 19(3):213–224
24. Duhovnik J, Tavčar J (1999) Concurrent engineering in real and virtual tool production. Concurr Eng Res Appl 7(1):67–79
25. Lida X, Zongbin L, Shancang L et al (2007) A decision support system for product design in concurrent engineering. Decis Support Syst 42(4):2029–2042
26. Raudberget D (2010) Practical applications of set-based concurrent engineering in industry. J Mech Eng 56:11:685–695
27. Avnet MS, Weigel AL (2010) An application of the design structure matrix to integrated concurrent engineering. Acta Astronaut 66(5–6):937–949
28. Bogus SM, Diekmann JE, Molenaar KR et al (2011) Simulation of overlapping design activities in concurrent engineering. J Constr Eng Manag, ASCE 137(11):950–957
29. Demoly F, Dutartre O, Yan XT et al (2013) Product relationships management enabler for concurrent engineering and product lifecycle management. Comput Indus 64(7):833–848
30. Tavčar J, Duhovnik J (2000) Typical models of product data integration in small and medium companies. Int J Adv Manuf Technol 16(10):748–758
31. Žavbi R, Benedičič J, Duhovnik J (2010) Use of mixed academic-industrial teams for new product development: delivering educational and industrial value. Int J Eng Educ 26(1):178–194

Chapter 23
Shipbuilding

Kazuo Hiekata and Matthias Grau

Abstract The shipbuilding process generally consists of concept and preliminary design, basic design, detailed design, production design and production. Design information is generated in each phase to shape products and operations in the shipyard. For each process the design activities are carried out with a high level of concurrency supported by various computer software systems, though quality of products and efficiency of the concurrent development process highly depend on experiences and insights of skilled experts. The detailed design information is difficult to be shared and design conflicts are solved in a common effort by design engineers in downstream design stages. Data sharing across design sections and simulation of the construction process to predict time and cost are the key factors for concurrent engineering (CE) in shipbuilding industry. The CE process in shipbuilding will be getting more and more accurate and efficient along with accumulation of design knowledge and simulation results. This chapter gives insight into the different phases of the shipbuilding product creation process and demonstrates practical usage through typical, comprehensive use cases from design and manufacturing. Finally, it draws some expected future directions for CE in shipbuilding.

Keywords Shipbuilding · Design and construction process · Data sharing · Design knowledge · Manufacturing process · Simulation

K. Hiekata (✉)
University of Tokyo, 5-1-5, Kashiwa-no-ha, Kashiwa 277-8563, Chiba, Japan
e-mail: hiekata@k.u-tokyo.ac.jp

M. Grau
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: matthias.grau@prostep.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_23

23.1 Introduction

In shipbuilding industry, the process of operation is very complex and a shipbuilding project from inquiry to delivery lasts quite long, about 2 years and more. The price of large tankers or bulk carriers is around tens of millions USD up to 100 million. Passenger ships, LNG carriers or offshore structures are even more expensive. A characteristic of shipbuilding is the huge volume of supplied material that needs to be procured and managed in addition to the design and construction process during shipbuilding projects. To present the importance of the concurrent engineering (CE) concept in shipbuilding, this chapter illustrates the ship building process first. Additionally, related works are reviewed to show problems many shipbuilders are confronted with. Moreover, several case studies are described in detail; an overview of future trends follows to conclude.

The basic shipbuilding operation is illustrated in Fig. 23.1. The detailed process can be found in [1]. The basic process is very similar to other manufacturing industries which produce products for individual customers. As for shipbuilding, two types of projects are considered for the design work. One is creating a new design model. The other is customizing past design models for the new requirements. Design and manufacturing information are required for both types of projects and often a design is reused together with the manufacturing information such as shop floor drawings, etc.

In concept and preliminary design, the basic specification is provided by the customer and the designers have to create a basic plan for the bidding. As the customer is interested in the cost for purchasing a vessel, the shipyard has to estimate the accurate cost for the delivery of the ship. A highly accurate estimation of the production is important for winning the bidding and improving the profit rate for the delivery of a ship. Skills of the estimation based on deep technical knowledge are required. Gathering data of past projects for simulations by commercial software systems are getting more and more important for accurately estimating the expected costs.

In preliminary design, designers work on key drawings such as general arrangement plan (GAP), Lines and Midship section drawings. GAP is a key drawing for defining basic dimensions, capacities and so on. A Lines drawing defines the hydrostatic performance by describing the shapes of the hull with curved surfaces. The Midship section drawing is a drawing for the most important part for the approval of the structural strength of the ship. Key performance parameters such as speed, fuel consumption, stability, basic structural plan, main engine and other key equipment are determined in addition to the three key drawings. Ship capability

Fig. 23.1 Standard design process of shipbuilding

and key performance are confirmed during the basic design process and a revised basic design will be provided for the contract. In the following detailed design phase, the detailed feature of the product is defined. As an example, the drawings developed in the detailed design phase could present handles for valves, small stiffeners, steel plates with curvature for the hull, and purchased products. There are not so many differences from other manufacturing industries in detailed design. One characteristic of shipbuilding might be that most of the parts for the ship hull and structures are made by cutting steel plates. Thus, a definition of standard parts is difficult. The number of parts is in the range of 100 k to one million for a ship, so this might also be a characteristic of shipbuilding. Depending on the construction process, the design model defined during the detailed design phase may or may not depend on the manufacturing facilities.

In production design, some of the drawings might be instructions for workers in shipyards or considerations for the manufacturing process such as margins. This phase may not include design trade-offs; this phase is a kind of planning for optimal manufacturing. The drawings do not only show shapes, dimensions and specifications of the parts, but also indicate how to make parts or fabricate assemblies. To construct a complete ship in dry docks, the whole ship hull is divided into building blocks to fit in the manufacturing facilities and capacities. Owing to the limitation of the manufacturing facilities, the production design can vary even for the same ship with the same detailed design model. The manufacturing information in shipbuilding considers the large deformation of steel structures by the welding process during fabrication.

In this section, the general shipbuilding process is described. Also, the characteristics and differences of shipbuilding are noted. To shorten the lead time, the whole process proceeds in a concurrent manner. A detailed structure or outfitting design cannot wait for the final design of the upstream process. Software systems for design and construction are implementing a lot of features and trying to provide integrated environments to facilitate the CE process; though they used to be standalone systems such as CAD or numerical control systems. To improve the efficiency of the shipbuilding process and handle the huge number of materials, PLM, ERP and more sophisticated software are more and more used by shipyards. There is a tendency to employ new integrated information systems in shipyards although the limitations arising from legacy design data and manufacturing facilities still exists.

23.2 Related Work

There are a lot of software systems supporting the shipbuilding process. The latest efforts for employing CE in shipbuilding are described here.

23.2.1 Problems in Scheduling in the Early Design Phase

The concept, preliminary and basic design phases are considered as early design stages. The literature for these phases will be shown here.

The purpose of the concept and preliminary design is to support the bidding process. Detailed information is not required during this design phase, nevertheless the shipyard should know the cost for the materials, man hours, major purchase equipment and the feasibility of the delivery date along with the on-going projects. The concept and preliminary design must meet the customer's requirements and, at the same time, have to be an optimal design solution for the shipyard in terms of constructability. The shipyard has to make a design proposal considering many trade-offs in the shipyard capabilities. Speeds, fuel consumption, hydrostatic performance, selection of main engine, strength of structure and construction weight are parts of the considerations in design. International rules of international maritime organizations and loading facilities in ports might be limitations for the design work. Even today, to achieve a balance in trade-offs, this phase of design process highly depends on human skills. Therefore, shipyards have to assign talented and capable people to the concept and preliminary design phases because these phases have a huge impact on total costs and schedule.

Meijer et al. [2] focuses on the pre-contract scheduling problem and captures the knowledge of experts for the process. Production scheduling tasks in the pre-contract phase are based on knowledge and experiences. The knowledge captured is, for example, detailed configurations of manufacturing facilities to optimize the turnover of the building dock.

23.2.2 Utilization of Engineering Software in the Early Design Phase

As just described, design engineers have to consider complex and concurrent processes of shipyards. The same situation can be seen in the subsequent basic design phase. To manage and predict the complex and concurrent shipbuilding process, basically two types of efforts are proposed for the early design stage.

The first approach is to accumulate design and construction experiences. In the early design phase, design engineers work based on similar projects. The designers identify the differences between the past design and the new requirements and estimate the impact on the new design.

The second approach is simulation. Production scheduling and performance measures of the ship (such as fuel consumption) are vital for bid creation whereas during the concept design phase the focus is set on production scheduling, ship performance is the key topic in the preliminary design phase. The basic design focuses on defining the parameters for the product to meet the requirements.

Though the trade-offs of design parameters across the design sections are taken into account in the prior stage, negotiations based on the actual design start in this phase.

NAPA facilitates the utilization of 3D design models in the preliminary design phase [3]. NAPA is a software company providing a suite of software for ship design and operation. The software suite for ship design covers the early stages of the design process, such as concept design, preliminary design and basic design. The design spiral in the early stage of ship design has a huge impact on overall performance and, furthermore, on the construction costs of the ship. Designers can easily elaborate the candidate for the basic design of ships using the NAPA software by varying some major design parameters. NAPA employs 3D models and the effect of changing the hull shape is calculated based on the current 3D design model. Complex interactions, such as hydrostatic performance and compartment plan, will be calculated in the software. Each software package employs many types of solvers in the basic design phase [4]. Computational fluid dynamics (CFD), evacuation simulation, structural analysis, vibration and acoustics are shown. Papanikolaou shows multi-objective optimization of a ship design case study [5]. This research does not consider the production process; however, a software tool for simultaneous evaluation of key measures is proposed and applied to a realistic case study. The simulation approach proved to be helpful for early design.

Integration efforts for CAD system and engineering software are also active. Bons has introduced the latest status of MARIN's software [6]. Hydrodynamic design tools are a kind of standalone software because of their specialized purpose. The integration of a third-party software framework enables specialized software tools for hydrodynamics to be applied to the early design stage. Ginnis integrates an in-house wave-resistance solver with CATIA to improve the efficiency to hull optimization [7]. As for structural design, Shibasaki utilizes a 3D design model for structural analysis within an early design stage [8]. The key is data conversion from CAD to a solver for structural analysis. There are many translators for data formats; however, the quality of the converted data is often not enough. It has been proven that, in an early design stage, a customized 3D design model can be reused for structural analysis with only few adaptions. The advantages of a large amount of design and production data for the downstream process have been illustrated by Nakao et al. [9]. According to their survey, quality and efficiency of the downstream process is improved if accurate design and production information is generated during the basic design phase. The research also notes that the proportion of man hour will shift from downstream to the basic design stage.

23.2.3 Collaboration Across Organizations in Detailed Design

Collaboration is important from the CE point of view [10–13]. Depending on the shipyard, hull structure and outfitting design units are working in the same area simultaneously. The structural design team doesn't want another team to make a hole for pipes, while the outfitting team needs that hole for an efficient routing of pipes or cables. This kind of design conflict is illustrated in Fig. 23.2. Some solutions of several major shipbuilding CAD systems follow.

AVEVA MARINE is a CAD system for shipbuilding software derived from Tribon Hull which was originally developed by Kockums Computer Systems, later called TRIBON Solutions. Now, AVEVA MARINE also employs the former AVEVA PDMS CAD system for 3D plant design. The software covers the entire ship design and construction process and also the integration of design and production. As for detailed design, design and process standards can be defined to fit the CAD system to each shipyard. Drawing and bills of materials (BOM) are automatically generated and collaboration across design sections is supported. These features help multiple design tasks such as structural design and pipe outfitting tasks to share design changes in detailed structures and changes of pipes and holes.

CATIA has been developed by Dassault Systèmes, and many automotive companies as well as a huge number of companies in many industrial domains use this software. Though the basic system of CATIA is a general platform, the system can be applied to the ship design process by using the feature for shipbuilding. With regard to shipbuilding, CATIA has a specific feature for pipe design. In the detailed design phase, the software can reserve a space for pipes without creating detailed models of pipes. The information, the desired route for the pipe design section, can be propagated in the data model without detailed design work such as checks for the design standard, or designing flanges, insulators, supports and other details. A route of pipes between the main set of equipment that is reserved in an early stage of detailed design is one of the advantages of the software. Some specific features for

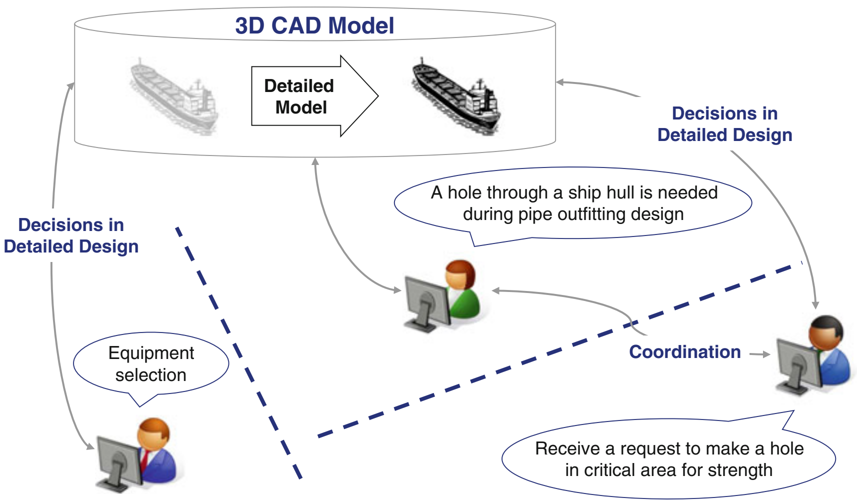


Fig. 23.2 Detailed design process with coordination across the design sections

shipbuilding such as a library for standard parts and parts generation by a macro are also supported in CATIA. In CATIA V4, ship hull design (SHD) is an extension to the product family to create hull structures of vessels. Starting with CATIA V5 and continued with V6 the hull structural design capability is covered by the structure functional design (SFD) module for the basic design phase and structure detailed design (SDD) for the detailed design phase.

FORAN is a 3D CAD system by SENER, Spain, and is meant for design and construction of ships and offshore structures. The software supports interactive piping design by checking and modifying features of the length of the pipes referring to the design standard and design review features for local rules of the shipyards defined in the system. The detailed latest feature is shown in [14]. Not only the interference between design sections, but also restrictions for bending machines or limitation of angles of elbows for material optimization can be considered for solving problems between detailed design and production.

23.2.4 Design Review in a Network

Reviewing the 3D model in a network enables the distributed team to work on the decision-making process along with the design progress. Sharing the updated 3D model is necessary to accelerate the speed of decision making and, thus, solve design and coordination problems. Ideally, all design work should go on concurrently, but the simultaneous update of the design model is difficult even with the deployment of 3D models. The number of design data and parameters is getting larger in 3D models. Moreover, 3D CAD systems usually work together with the traditional in-house software, while the complexity of design practice is getting higher and higher. From the information systems' point of view, just light and sufficient design data should be transferred to designers distributed in the network. However, the question, which data is important and necessary for which point of design process remains difficult. There is no answer still, while most of the software provides features for exporting light-weight models formatted in basic standard 3D format (see Chap. 11).

Collaboration based on sharing 3D models in a network is achieved by sending small sets of data required in design reviews, not by sending the complete data of the design process. For example, the software technology at the client's side is a standard rendering system based on OpenGL. A standard data format such as XGL in XML only delivers shapes and dimensions for an efficient data transfer. The detailed data for other parts can be stored as metadata to leave the handling process of detailed attributes to the generic database system. In general, software features for sharing 3D models are developed based on the open standard.

The JT format proposed by Siemens PLM and published under ISO introduces a method for sharing design data across CAD systems capable of handling this format (see Chap. 11). Especially for sharing rich design models in a CAD system-independent way, 3DPDF developed by the 3DPDF consortium as ISO standard is an alternative for collaboration across companies [15]. The format includes exact as well as light-weight shapes, metadata, dimensions and also product manufacturing information (PMI) such as tolerances (see Chap. 11).

NUPAS CADMATIC is a joint-venture product of Numeriek Centrum Groningen B.V. and Cadmatic Oy. These two companies are specialized in ship hull and steel part constructions and ship outfitting respectively. NUPAS CADMATIC provides a collaboration platform via internet. The platform works as a data server, while the 3D models can be shared. Light-weight data models can be sent by normal e-mails and will be reviewed by eBrowser, their free software for design reviews. Reviewers can give feedback in the software as other CAD systems do as well.

The schematic of the light-weight data shared in most of the CAD systems for shipbuilding is illustrated in Fig. 23.3.

23.2.5 Knowledge Management

Automatic check features for designs are useful for keeping the quality of design high. In order to allow automatic checks, know-how and design rules need to be stored in the CAD systems. The know-how, knowledge, and standards stored in the

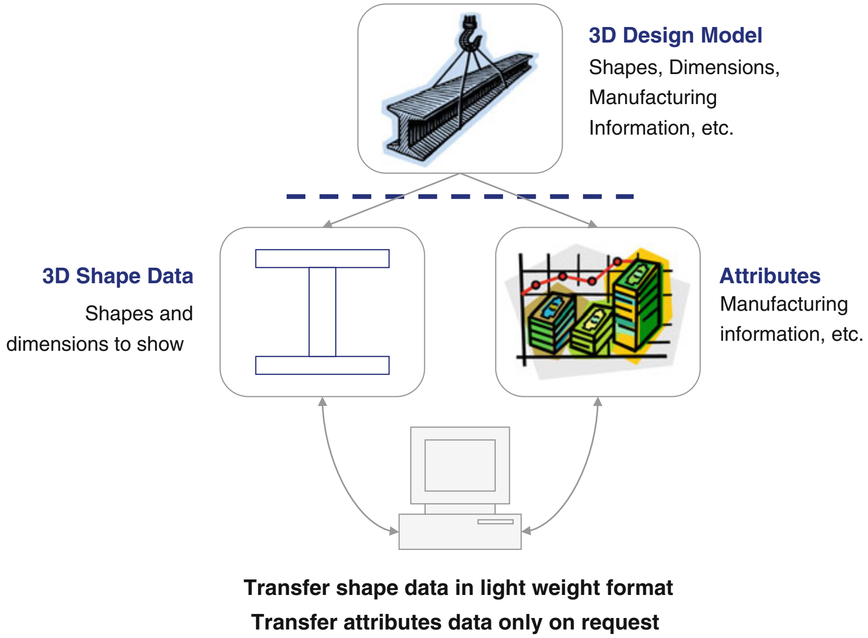


Fig. 23.3 Schematic for design information sharing

systems are, for example, the size of passage space, limitation of gradients for drain pipes, accessibility for maintenance of equipment and so on. These tips can be accumulated by means of the software systems. The difficulty of accumulation of know-how and hints is known as the knowledge acquisition bottleneck in former studies. One solution might be an extraction of the rules from text data generated during daily operations in shipyards [16]. Several other practical solutions have been proposed (see Chap. 10). The rules and knowledge accumulated should be managed well to improve future designs [17].

The software feature to accumulate know-how and rules is mentioned in this section [18]. If knowhow and rules are stored in the systems, those are also helpful for learning design knowledge [19]. Basic and routine checks should be automated by software systems and design engineers should focus on learning from accumulated knowledge and maintaining the knowledge (see Chap. 10) [20].

23.2.6 Integration with External Systems

The deployment of new software systems to shipyards also should be called a kind of integration rather than only development or customization. Similarly to general CAD systems, a shipbuilding CAD system is also required to be working together with many external systems. In shipyards, there are many types of software systems running. In downstream design work, such as detailed design and production design, the information on delivery dates or prices of the parts are helpful in addition to physical shapes and dimensions in BOM systems because the installation of purchased equipment completely depends on the delivery date.

One simple scenario of working with an external ERP system is shown in Fig. 23.4. Design and production data are handled by the CAD system. Broader information is stored in the PDM system, while the ERP system handles inventory, procurement, finance and others.

There are many efforts to implement ERP in shipyards, as well as to apply PLM. Larkins has worked on the development of a neutral data format for shipyards and integrated CAD and ERP [21], based on the ShipConstructor product of SSI. ShipConstructor is a shipbuilding 3D product modeling software running on top of the widely used AutoCAD system. All shapes and attributes are stored in a Microsoft SQL server which makes it more reliable and enables concurrent design. Lin and Gonzalez point out that integration of CAD, PLM and ERP in shipbuilding should be CAD oriented [22, 23], while Rong recommends the utilization of cloud storage for CAD/CAM and ERP integration [24].

Many CAD vendors recommend CAD oriented data integration to fit the current process. Many of the systems focus on linking up the whole process and data, which then means including the purchasing department, the hull structural design and outfitting departments and design work across the design and construction departments. The concept of the integration is the lifecycle of the ship production as supported by PLM systems [25].

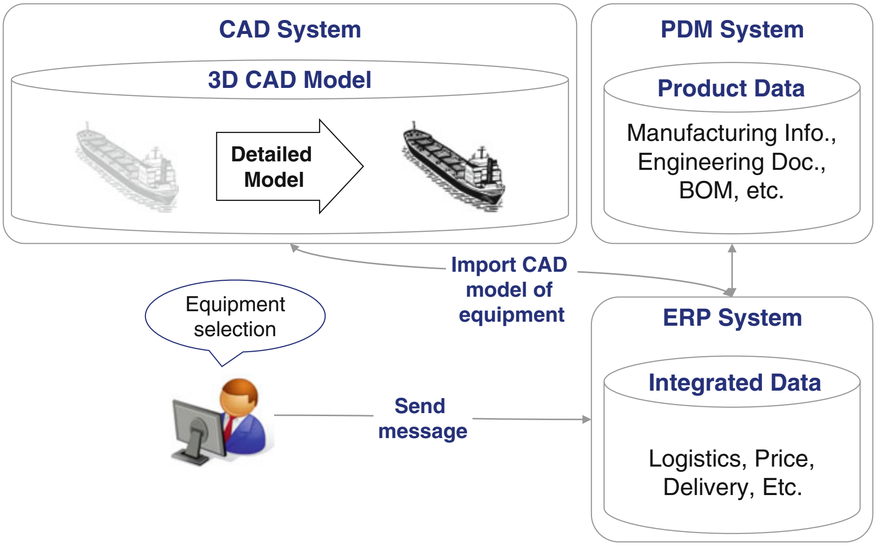


Fig. 23.4 Interaction of design work in CAD, PDM and ERP system

23.2.7 Considerations for Production

One consideration concerning production design is making an efficient plan for welding in the upstream sector. During the fabrication process, a lot of man hours are needed for welding. The environment for the welding processes such as upward has impact on the production costs. Assemblies of ships are huge and cranes are needed to turn the assemblies. The capacities of the cranes, distance of the areas in the shipyard and the weight of sub-assemblies should be considered to optimize production cost.

Even if the design (or the generated 3D model) is not yet completed, an optimized production procedure is required in advance. Park works on the management of deformations by heat to reduce rework [26]. The deformation is predicted by a solver and the results of analysis give feedback regarding the design model. The production will proceed based on the production design. Basically, the procedure as defined in the production design phase will be completed during production. Storch ran simulations of the production process to predict productivity by changing the work environment [27, 28]. Simulation is a powerful tool to improve the production design's quality and efficiency of the production.

23.3 Case Studies

To demonstrate both design and production of ships, we have chosen several “use cases” per domain. Interoperability between design tools as well as seamless supplier integration into the design process facilitates successful engineering collaboration [29] (see Chap. 7). Simulation for predicting the performance of the final product and the efficiency of production is a crucial point of related works. Several case studies illustrate details of the simulation technique.

23.3.1 Design of Equipment and Outfitting

Using the right toolset to design and manufacture outfitting is of paramount importance and should cover all needs from the design of outfitting structures like fundaments for layout and routing task for electrical and hydraulic components up to support for process planning and numerically controlled manufacturing.

The “best-in-class” approach uses the best available tool per design discipline, e.g., one for outfitting structures design, one for piping, etc. The benefit of this approach is perfect support in a discipline with features typically not available in the common denominator. The downside of this approach is the need to integrate the separate tools into a common toolset. Care needs to be taken that the necessary exchange of data from and to the distinct tools does not render the benefits in the various disciplines useless.

Blohm and Voss Naval (now Thyssen Krupp Marine Systems), a German shipyard with a long track record, have performed a project for investigating the ≪best in class≫ approach with the mechanical CAD system Siemens PLM NX for outfitting and piping design. This system offers a flexible solution to the design of mechanical structures created from sheet metal and profiles up to the placing of components and routing pipelines. Design of ship hull structures, and work preparation including plate nesting and creation of NC files for profiles and plate parts on the other hand was done using AVEVA’s TRIBON M3. This necessitates a powerful interoperability link to transfer the manufacturing data from NX to TRIBON as an alternative to repeating manual rework [30].

The most important findings during the investigation of requirements was that designers could easily create parts in NX that neither obey the yard standards nor were supported by manufacturing. As a result this requires performing the appropriate customization of NX and some kind of validation functionality as part of the link between two CAD systems. The customization of the CAD system serves two purposes: it should support designers with ready-made building blocks like profile cross sections and it should ensure successful data export. As with all kinds of customizations it is important to find the right granularity of building blocks (Fig. 23.5). There should be a balance between patronizing the designer by providing only a few canned solutions and leaving too much freedom.

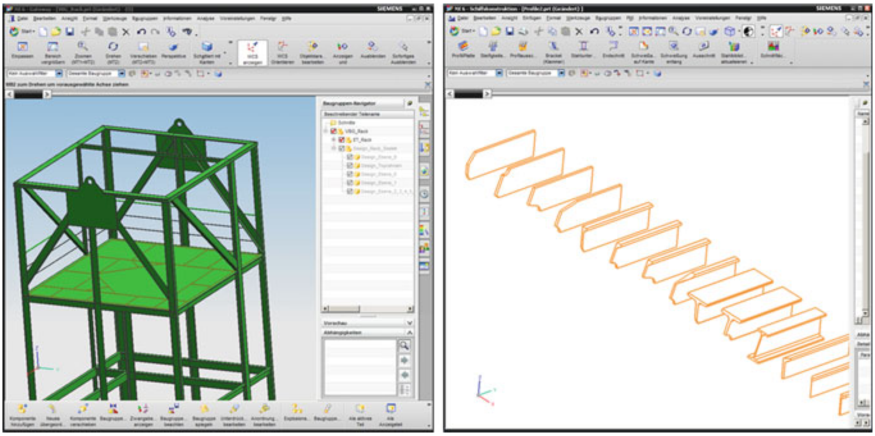


Fig. 23.5 Example structures and profile customization

The second and more involving step in the project was the design and implementation of a solution for the link between the two CAD systems. This link had to transfer the information created in NX containing the manufacturing geometry, validate it against rules defined by the yard and import it into TRIBON for manufacturing purposes. The fundament for all involved activities is the data model of the link solution.

Experience suggests defining a data model not too closely tied to those of the source or target system. It is rather a representation of the complete business data and does not rely on implicit knowledge available only in either of the linked CAD systems. This has the benefit that additional source or target systems are feasible without too much hassle. Furthermore, this “link model” is created in the spirit of the model-driven approach and uses standard technologies like XML schema and JAXB data binding (Fig. 23.6). This is essential as the project software is handed over to the yard. Using standard and openly available technology avoids a vendor lock in and enables further maintenance and development by the customer.

Another aspect important for the daily usage is the validation of the parts designed in the CAD system. While it is common practice to incorporate yard standards to a certain degree into the customization, it is neither useful nor technically feasible to prevent all kind of errors that way. Especially if designers work on multiple projects with different standards it is easier to catch the corner cases by a separate validation step than to switch CAD system customization every here and now. To support this need the link implementation contains a dedicated rule engine used to enforce rules like

- validate the combination of material quality, thickness and dimension for profiles and plate parts
- validate the combination of profile cross section and endcut types
- validate dimensions for endcuts
- validate naming and numbering of items

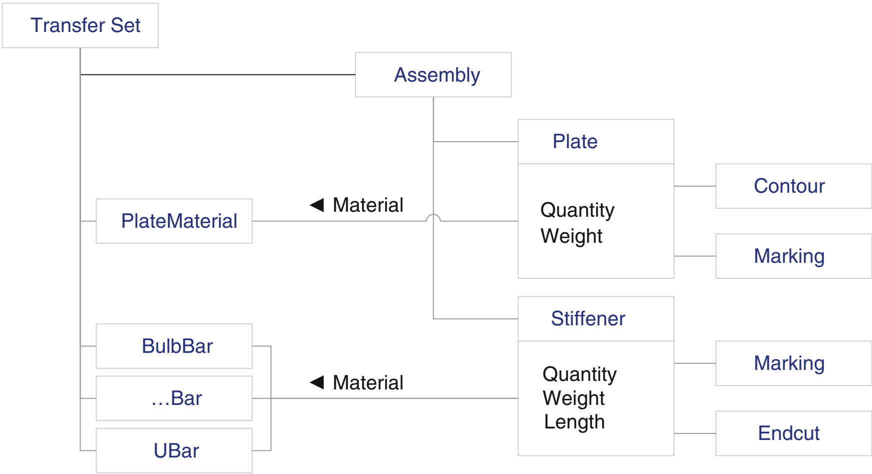


Fig. 23.6 The neutral XML format used as internal format allows easy extension

The rules engine supports checks on all kind of properties using allowed ranges, enumeration or regular expressions as well as conditional checks. The underlying rules are not hardcoded within the software but read from dedicated validation files. These files contain the rules in a domain specific language (DSL) using the vocabulary of assemblies, plates, etc., to allow changes by an administrator at the yard. A rule violation stops further processing and the validation results are shown to the designer. For this very reason the link is implemented as GUI application as the direct feedback enables the designer to change the problematic parts.

Using a “best-in-class” approach is a viable solution to support the design of outfitting parts. However, certain attention is needed to integrate the tools and gain a continuous data flow. In this solution the integration is provided by customization of the source system and tailor made link implementation. It does not only transfer the data into existing systems but also performs project specific quality checks and avoids later processing of non-compliant data. Using such a link enables the yard to use commercial off-the-shelf software without breaking existing business processes harnessing again all the benefits of dedicated “best-in-class” solutions. Furthermore, this approach facilitates continuous process improvements on singular steps in the entire process chain.

23.3.2 Collaboration Enabled by Intelligent PDF Documents

Portable document format (PDF) is a ubiquitous and widely supported document format with Adobe Reader® found on most every personal computer. Over the last several years, functionalities inside PDF have evolved to better support the engineering communication processes including built-in capability for the rendering of 3D CAD in combination with many features found traditionally only in dedicated CAD Viewers [31]. PDF capabilities include such features as digital rights management [32] (see Chap. 18), an ability to include data from throughout the enterprise, 3D CAD visualization, commenting, markup, measurements and PMI (see Sect. 11.3.2). PDF documents can be used for the communication between various stakeholders. In the engineering domain, especially review and approval processes can be supported very efficiently.

23.3.2.1 Layout and Planning Information in Early Development Phases

In early phases of shipbuilding processes, layout and planning information has to be exchanged between different partners in the shipbuilding supply chain. An example for this is the planning of a ship's machinery room. The main engine supplier has to provide information to the shipyard, which gives an overview of the main dimensions of the engine as well as of requirements for additional space, which is needed for maintenance and service tasks.

Until now, this information is typically communicated by drawings (either on paper or as scanned documents). These drawings are often very complex to be understood by an "external" partner, so the use is time consuming and error prone.

With 3D PDF, this information can be communicated on a level, which is very easy-to-use on the one hand but on the other hand adds additional functionalities into the PDF document by including active 3D geometry combined with PMI (e.g. dimensions or tolerances) into an interactive document.

23.3.2.2 Web Based Assembly of Multi 3D CAD Data

Another use case for 3D PDF is the assembly of CAx data coming either from different CAD systems and/or from different partners. With a web based structure browser ("Interactive Assembler"), different CAD assemblies or parts of them can be selected and afterwards automatically converted into one PDF document. This document then includes all 3D content together with all structure information from the original CAx applications. This PDF document can be used for review, lightweight collaboration or communication processes (Fig. 23.7).

The end user can create a simple and easy-to-use assembly of CAD data from different sources without the need for any CAD seat or an additional viewing tool. Through PDF, CAD independent 3D viewing now can be done on any workstation worldwide. Of course, the functionality is limited and only addresses end users without the need for changing the CAD data themselves. But any user who only has requirements for viewing and checking CAD data, especially outside engineering or along the supply chain, can use these new capabilities without any additional tools or infrastructure on the client side [33].

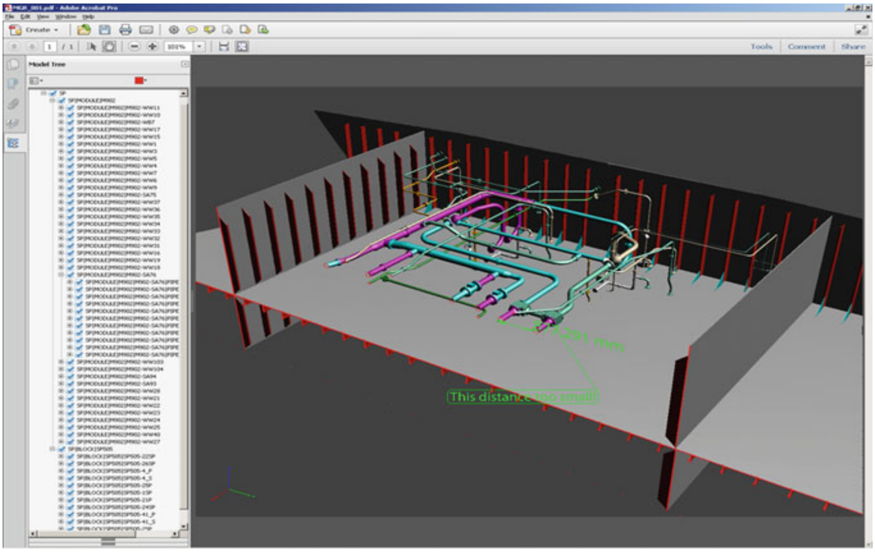


Fig. 23.7 3D PDF document with embedded active 3D CAD geometry (cross sectioned)

23.3.2.3 Cross Enterprise Review and Change Management Processes

The PDF documents can be used for communication between different participants in an approval or change management process which can be long and expensive (Fig. 23.8). Each participant can add comments or redlining into a PDF document (Fig. 23.7). Besides, the document can be approved by adding a digital signature into the document. In combination with rights management a lifecycle can so be built into a PDF document allowing a controlled circulation without having control over the document itself.

Most importantly, the added content of the PDF document (comments, annotations, form fields) later on can be exported again from the PDF document and stored back into virtually any enterprise system. Or, as an alternative, all comments coming back from various participants of a review process could be aggregated into one PDF document. This document then gives an overview of all comments from various review participants in one document.

23.3.2.4 Manufacturing Documentation

Manufacturing documentation means all documentation used for manufacturing processes of a product. This includes work instructions, which are used by workers to assemble products by performing a number of manufacturing steps as described in the instruction. Adding 3D geometry into such documentation avoids long text

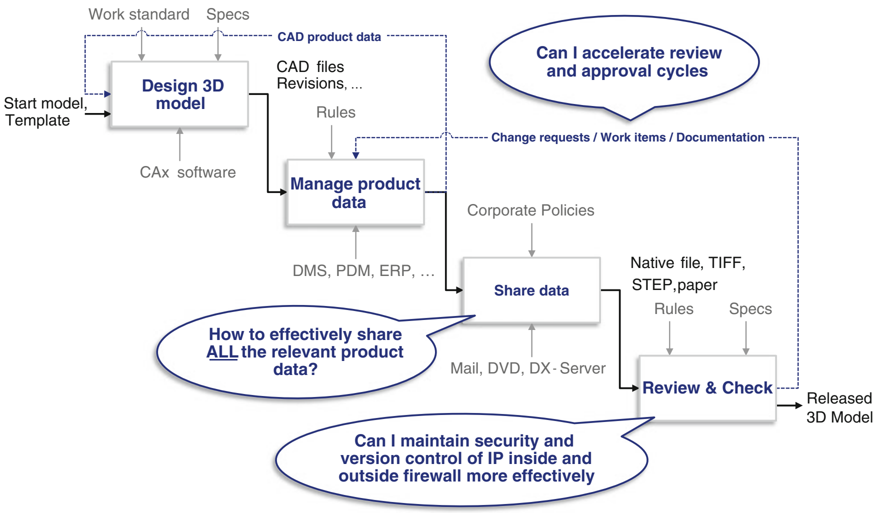


Fig. 23.8 Change management with 3D CAD data

sections and also makes it easier for a worker to understand how a manufacturing process has to be executed.

An example from a recent project is the documentation of holes to route pipes through steel structure. The process is triggered by the piping designer, who selects assemblies and parts to be exported into a 3D PDF document. An automatic process for the generation of a 3D PDF document is started. Together with geometric information (which, of course, also includes the representation of the requested holes), additional metadata information for each hole is exported.

In the PDF document, a 3D representation plus a list with the metadata for each hole is included. The content types in the document are interlinked, so the receiving steel designer can click on the metadata of a specific hole which directly links to the 3D representation of this hole in the context of the steel structure in highlighted mode. By directly showing the active 3D geometry of a ship's section with all of the holes to be included, the steel designer gets appropriate information for his task to approve the requested holes in a very efficient way. He can work on a plain workstation with Adobe Reader and does not need any CAD or PLM access at all.

A similar use case form the manufacturing domain is the "illustrated BOM". In this use case, PDF brings together the BOM coming from an ERP/PDM system and the 3D geometry coming from a CAD system. This information is linked together in the PDF document, meaning that the user in manufacturing can click on an item of the BOM list and automatically gets linked to the 3D view of the selected part(s) on his screen.

23.3.2.5 Supply Chain Collaboration

All aforementioned scenarios imply a point-to-point connection between two partners (e.g., shipyard and supplier) who are familiar with each other. In case of a shipyard which has to keep relationships with many suppliers simultaneously, manual management of data exchange becomes too complex and has to be maintained through specific tools. Furthermore, data exchange with partners in the product development process requires some additional topics to be addressed:

- **Security**: Public networks are open to everybody—sensitive information needs to be exchanged in a secure way.
- **Reliability**: Public networks are often not as stable as required, especially for transmission of large amounts of information (e.g. large CAD model file packages).
- **Traceability**: For the exchange in a globalized economic environment often—even legally binding—prove of data transmission and reception is required.
- **Efficiency**: Process security (such as repeatable exchanges and defined content) without loss of competitiveness becomes more and more crucial.

The speed of bulk data exchange not only depends on the available network bandwidth, but also on network latency over lengthy transmission distances. As soon as distances are long and data volume is high, latency can result in considerable delays and low throughput. This calls for a parallelization of the transmission stream as known from so called peer-to-peer networks. Such networks move single files in multiple threads in parallel, thus distributing data transmission on several channels thereby achieving a twice to three times better throughput. In cases where the connection is lost despite all efforts to minimize latency, a mechanism to resume data transmission where it left off is required. Records of who has uploaded or downloaded what data and when are kept via appropriate logging functions, so that they can be traced at any time.

Document management and provision features are needed, so a capability to organize exchanged files in a structured way (e.g., a by project and/or by exchange partner) will be required. In addition a publish/subscribe mechanism will be helpful in order to keep recipients informed about new or updated information. Updated files need to be managed by the exchange platform in a way that they are only available for download in their latest version and outdated versions are optionally archived.

Given the vast quantities of data and the numerous exchange processes handled by many companies on a daily basis, users should not be expected either to upload their data to the exchange portal or to download incoming data manually. A capability would be needed to automate these activities thus synchronizing the incoming files at the exchange platform with the local file system of the recipient. Even more, this should not only be available on a user basis but needs to be part of the companies IT infrastructure to avoid administration effort grows linear with the number of users.

Where appropriate a capability is needed to integrate the exchange process into back-end systems such as PDM on the sending and/or receiving side of the exchange. This allows locating or dropping all the relevant information in the internal data management environment thereby bypassing an intermediate storage in a file system and even enables automated further processing to take place such as creating a revision or locking.

All these requirements can be implemented by a secure data exchange platform which represents the hierarchy of the entire supply chain network (Fig. 23.9). The basic communication runs via web, the tool is installed in the demilitarized zone (DMZ) and can thus be accessed by all involved parties.

Prior to its implementation in shipbuilding, this solution was already widely used in other industries [34] which employ more than 10,000 users per installation. Usually it is included into supplier portals as an additional feature. For each partner, of which the number can be almost infinite, a specific exclusive web-space can be allocated which allows encrypted data exchange via the web. For processing of data workflows several functions can be defined, e.g., encryption/decryption, translation, filtering, quality check, packaging, check in/out to databases. On the shipyard site user-specific pre-processing dialog and methods can be defined, e.g., connecting the Adobe Life-Cycle product suite for derivation of 3D PDF models with corresponding PDF sheets and presetting the access rights to singular documents. In the solution a role and access concept is stored as well which can seamlessly interoperate with the Adobe suite.

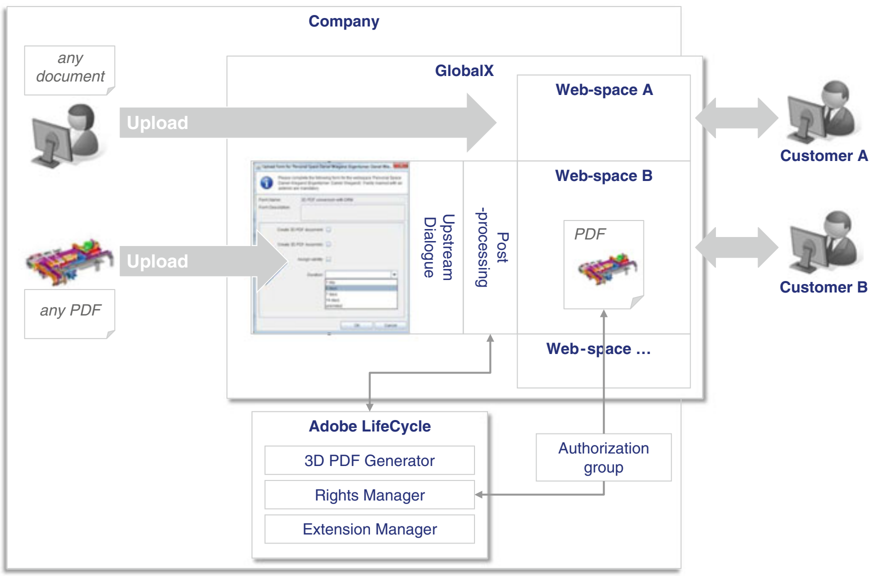


Fig. 23.9 Supply chain collaboration enabled by managed file transfer

The solution allows the distribution of data packages to multiple receivers (e.g., in case of request for quotation). Subscription of specific content (e.g. project relevant CAD models) is also possible. After the receiver has downloaded the data package, the sender gets the corresponding message and, thus, is aware that the data exchange has been executed properly. The user has to select the data for the exchange, his exchange partner (receiver) and the processing method (e.g. native CAD data, exchange as PDF package). After starting the action the entire processing runs fully automatically.

23.3.3 Data Management in Production Process

Recently, some accuracy evaluation systems using measured data of assemblies obtained from laser scanners are proposed. Laser scanners measure the whole surface of the parts as point cloud data. Measured data can be used for evaluation, checking the accuracy of shipbuilding blocks [35] or surfaces of shell plates [36]. The measured data and evaluation results have much information content, so these data are expected to help to discover knowledge about the manufacturing process. However, in most shipyards, the search and reuse of evaluation results is difficult because large amounts of accuracy information are stored without an adequate data management. To utilize the accumulated data, a data management system for measurement data and accuracy evaluation results of shipbuilding assemblies gauged in the manufacturing process is needed.

The proposed system has three functions: (1) accuracy evaluation, (2) accuracy data accumulation, and (3) search and reuse of accuracy data. The objective of the study shown in this section is to build a method for identifying knowledge, know-how and techniques in the field based on the data managed by the developed system and evaluated by the three dimensional measured data in the ship construction process. The overview of the whole system is shown in Fig. 23.10. All types of data are stored in the database and, as well, the metadata is assigned to the data. Any data stored in the system can be reachable efficiently thanks to the metadata.

In an accuracy evaluation system, the accuracy of assemblies is calculated by comparing measurement data obtained by a laser scanner to design data. The methodologies for an accuracy evaluation are different according to assembles, and some existing method can be applied for the evaluation.

In the accuracy data accumulation system, measured data, design data and evaluation results are accumulated according to name, feature, or evaluation result of assembles. The metadata is attached in resource description framework (RDF) format [37] and has URI for identifying the accumulated data. Relationships of each assemble are structured in RDF format, and the user can edit the relationships.

In the accuracy data search system, data are searched by querying RDF metadata attached to accuracy data. The value of the attached metadata and the name of assemblies defined in the RDF relationships visualized as a tree structure are used for metadata search with SPARQL. Search results are displayed as a summary of

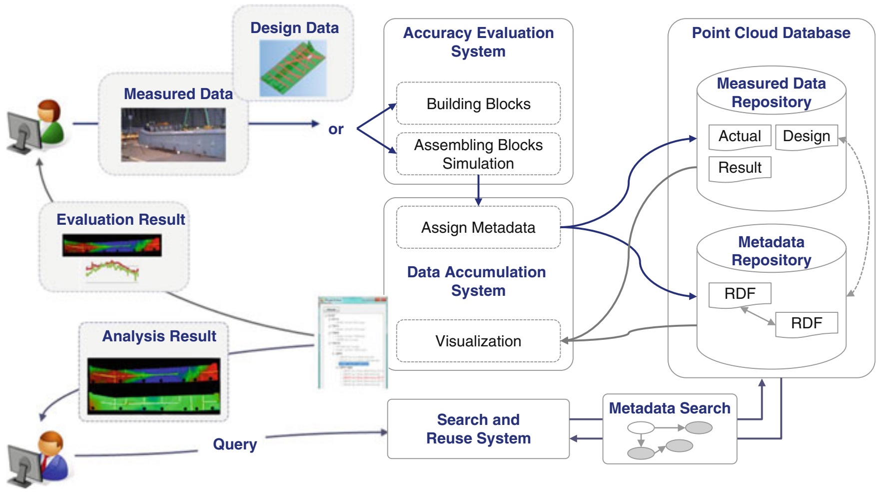


Fig. 23.10 Overview of the data management system

metadata, while accuracy data identified by the searched metadata are loaded and compared.

The accuracy of sub-assembly parts manufactured in a shipyard is evaluated and measured by data accumulated in the proposed system. In these experiments, a decrease of distortion of the panel surface is confirmed by comparing the accuracy of the panel before and after the heating process. This system is also helpful to identify areas featuring high distortion by searching data extracted from measured data and comparing it to the evaluation result. The findings obtained by these comparisons can be utilized for redesign of the manufacturing process.

One result of case studies with the system is shown in Fig. 23.11. This figure gives an overview of the shipbuilding blocks and the deformation of internal structural members calculated from accumulated measured data by laser scanners. The vertical axis of the graph is offset along with the depth of blocks and the horizontal axis corresponds to the width. Two measured data are retrieved in a 3 months interval, however the same tendency of the deformation can be found.

The accumulation of the data will enable shipyards to do this kind of analysis easier and avoid uncertainties in the production process. The accuracy of shipbuilding blocks can be evaluated by the proposed system. The deformation of the shipbuilding block in the production process will be recorded by the raw measured data and the analyzed results. The evaluation of the feasibility of the system is going on.

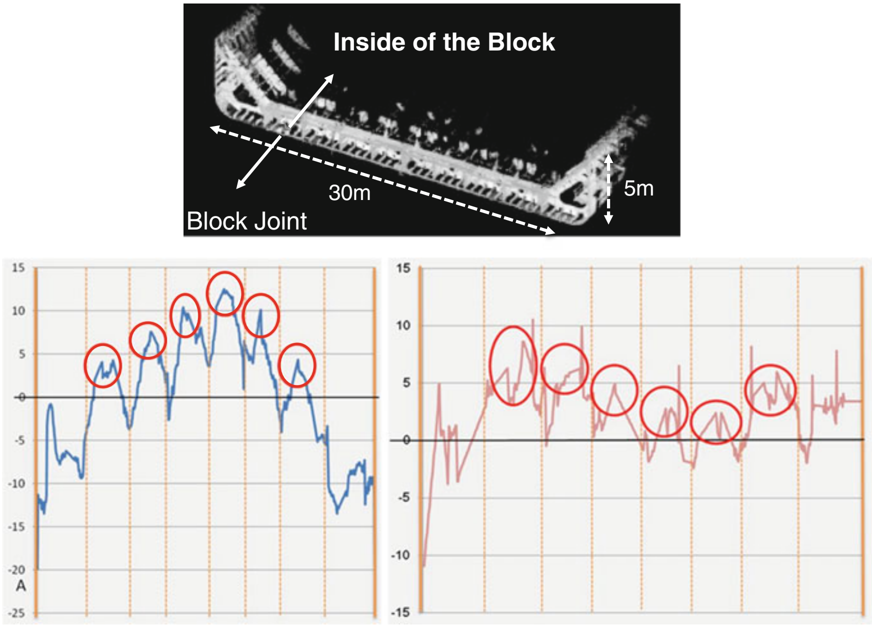


Fig. 23.11 Deformation in fabrication process

23.3.4 Simulation of the Production Process

As described in this chapter, simulation techniques are crucial to predict the subsequent process. Especially the process which requires huge volume of man hour such as production process should be correctly estimated. This case study proposes a methodology to evaluate organizational performance based on the research described in [38].

The developed system defines workers, facilities, activity models and a production strategy. The evaluation of organizational performance is done through the following processes: (1) create the enterprise model and strategy, (2) calculate a work plan by optimizing the weights of each strategy, (3) compare the basic scenario to the scenario of a changing situation. The system proposes an initial work plan. The plan minimizes the total cost in doing the work activities considering the weight of each production strategy by introducing genetic algorithm.

The proposed methodology is applied to some sample scenarios in a fabrication shop. Results show that the methodology can evaluate organizational performance successfully by analyzing the work plan. In addition, the methodology also evaluates the effect of improving organization and sudden trouble quantitatively. Figure 23.12 shows the overview of the proposed method.

Initially, an enterprise model is developed based on the workers and facilities in an organization including the different work activities and skills set, while the

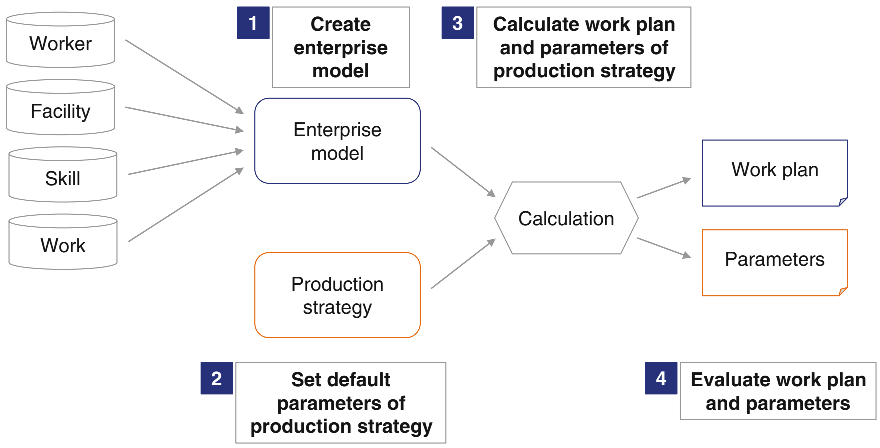


Fig. 23.12 Overview of simulation system for fabrication workshop

production strategy is made by setting each parameter. Next, the optimal work plan for the enterprise model is calculated by designing the parameters of the production strategy. Finally, the organizational performance is examined by evaluating the optimal work plan and parameters of the production strategy.

The skill set is a class of skills needed to perform the various activities in an organization. Workers, facilities and tasks in some activities are defined by the skills in this set. The organization model is composed of workers, facilities and their capabilities or skills. Workers and facilities are defined by their costs and the presence or absence of skills in the set.

This method is evaluated in the fabrication process of simple panel structures in the case study. The process model is shown in Fig. 23.13. The simulation scenario is that 11 workers using 6 facilities are working on making 10 panels. The result is also shown in Fig. 23.14 in Gantt chart format. The weight vector for the strategy for assigning activities to workers is also obtained. This simulator shows that the job allocation strategy will change from cost saving to first-in first-out to keep the delivery date in case of resource shortage. The simulation results are suggestive however the effort for making process and organization models is barrier to actual deployment to practical situation.

23.3.5 Summary

The first case study demonstrates advantages of using “best-in-class” design tools combined with powerful interfaces. These advantages comprise gaining benefits of high productivity and high user confidence.

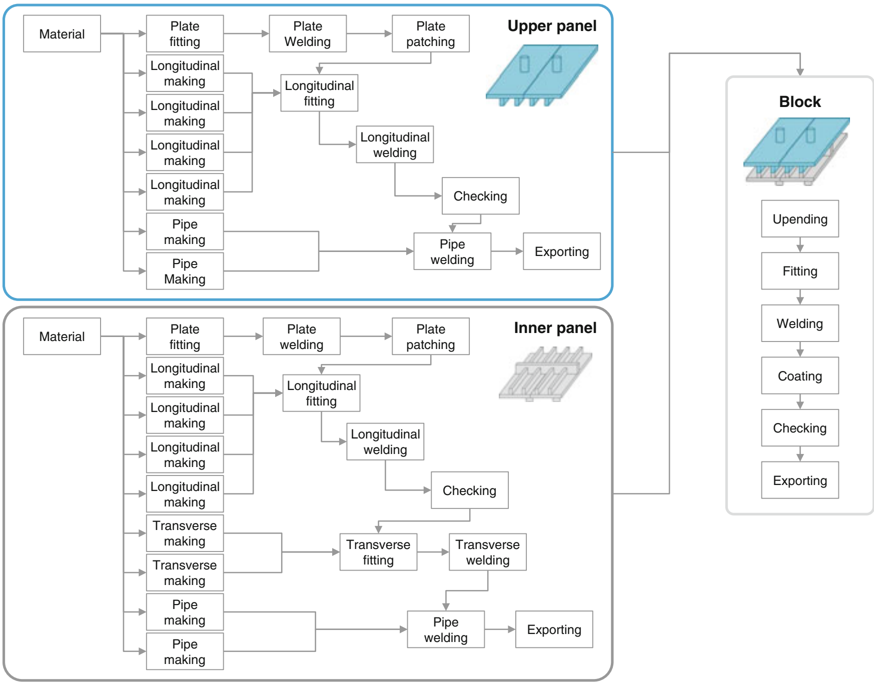


Fig. 23.13 Process model for fabrication of simple panel structure

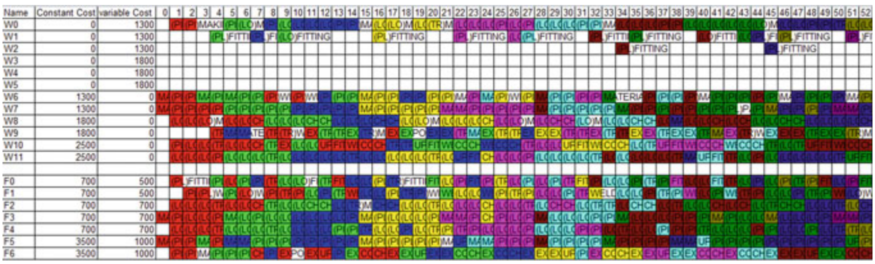


Fig. 23.14 Calculated schedule by proposed system

The second case study with multiple scenarios in engineering collaboration shows the benefits of using intelligent documents with embedded 3D data implemented in 3D PDF.

The third case study shows the power of accumulated data to improve the current practice. There is a plenty of data in manufacturing practice, however, the utilization for improvement is still emerging. The data can be called big data and its utilization in future is expected.

In the fourth case study, tips for scheduling can be found even in a simple process model. Simulation results depict the details of operations, making the simulation very helpful for engineers to make decisions for planning. The detailed behaviour of complex and concurrent downstream process can be simulated to reduce the uncertainties in the plan.

23.4 Future Directions for CE in Shipbuilding

Direction for the future in shipbuilding is discussed in this section. Challenges to improve the design process and the state of the art of the software feature are discussed.

23.4.1 Early Design for Concurrent Engineering

Shipbuilding industries spend a lot of time and cost on fabrication and assembling work. In addition, the volume of man hour in production is subject to the design phase. In other words, upstream design has a strong influence on the time and costs of production. The costs for production can be reduced by defining more accurate and detailed design parameters in the upstream design process. In practice, uncertainties in production can be avoided by employing detailed 3D models into the design work to fix more design parameters during the basic design phase. Concurrency and flexibility of the design process are required to improve the plan. Upstream design should resolve design uncertainties as far as possible to facilitate the CE design process in downstream.

Without 3D CAD technology, fewer design parameters can be considered explicitly in the basic design phase, and conflicts caused by the limitation of the design parameters are solved only in a downstream design process such as the detailed design and production design phase. Design experts require skills of many implicit design parameters and considerations. Designers can learn these skills in on-the-job training and from their experiences. The deployment of a 3D CAD system and the improvement of its computational power will enable description of a complete ship model containing more than half a million product parts and simulation of the process to assemble all the parts. Model and process can be simulated in detail by computers, and taken into consideration.

Design systems supporting the management of the CE approach are expected to significantly improve design quality in shipbuilding. This can be an innovation for the design process. The decisions in each design phase are taken to the subsequent process as design information. The next process assumes the decisions of the preceding process as design constraints and defines more detailed design parameters. The time history of design information generated in the design process is illustrated in Fig. 23.15. Considerable time and costs are spent on the creation of

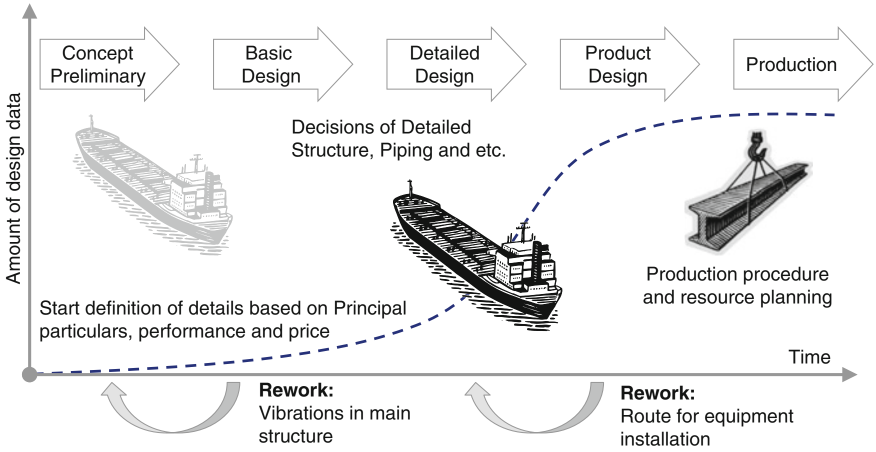


Fig. 23.15 Design information generated along with the design process

detailed models and drawings by defining many dimensions and attributes of parts. Although the number of design parameters and man hours in design work are not that big within the basic design phase, significant time and costs are spent for the design work based on the results of the basic design process. If design problems or conflicts caused in basic design cannot be solved within the downstream process, the rework of the problem will have a very serious impact on the whole process. Predicting detailed operation of the following process and efficiency of working on the downstream process can have a positive impact on the shipbuilding industry.

During the detailed design process, quality and quantity of design information within an early design stage can reduce time and costs for design more than the effort of resolving design conflicts downstream.

The quality of the concurrent process can be improved by prediction; using 3D models and simulation techniques will play a key role [39]. For example, if there might be plans to build block divisions for production, the trade-off of each plan can be shown with quantitative data and the best plan can be chosen based on costs, schedule and other limitations. Time history of design information will change as shown in Fig. 23.16. Much more design information will be created within an early stage of the design process. As for designing simple bulk carriers, more than 20,000 h are spent on the design work and the efficiency varies according to the quality of work in the design process. The concept of CE was introduced to shipbuilding industry years ago and has delivered improvement; however, barriers for implementing fully concurrent and front-loaded design, such as local optimization and data conversion, still exist [18, 20, 39].

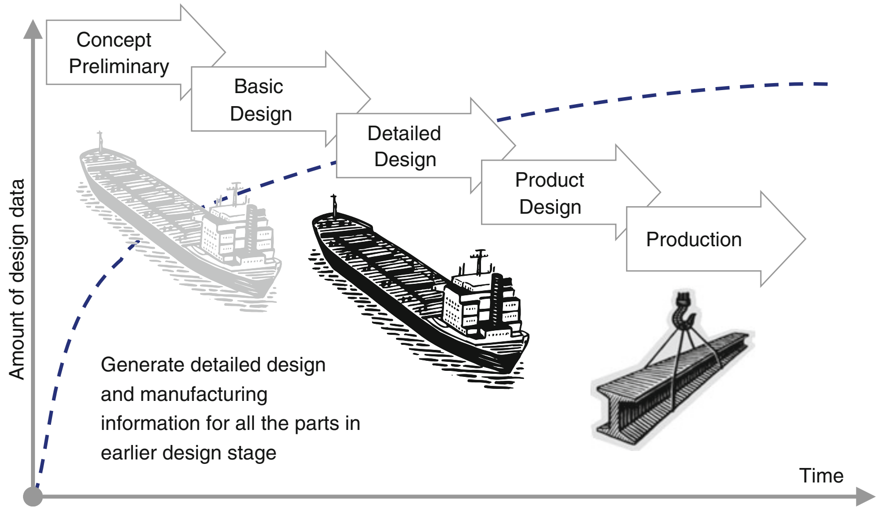


Fig. 23.16 Design information generated along with the concurrent design process

23.4.2 Reflections on the Organization Structure

The information infrastructure of CE is still not elaborated enough for shipbuilding. The design process is going on concurrently but the data is not shared in real-time because of the limitations of the information system. Consequently, a design conflict will emerge after combining design data. The route of pipes and cables is often blocked by structural parts, while usually the hull structural design section has priority over outfitting design sections. This problem is solved by some work-arounds with extra man hours in production design or during the production phase. Early integration of data can reveal this problem.

The concept to improve the design process is very simple, but time and costs of defining detailed 3D models in basic design will cause a large increase in man hours within the basic design phase. To make good use of the CE concept, in addition to the support by a sophisticated software platform, shipbuilders are required to change the organization of the design department to have a larger number of engineers in basic design process.

23.4.3 Concurrent Engineering for Production

Ship design and construction process are going on in a concurrent manner even in a very early stage of the process. Some problems exist, however. For example, shipyards need to purchase steel for the structure in an early stage of design, while the detailed structure has not been defined yet and the exact amount of steel is unclear. For a bidding process, the costs of production should be known, though the weld length cannot be known without a detailed structure. The accuracy of estimation depends on the experiences of designers. Human skills are thus needed to make the process concurrent.

To start the production process without the final design, 2D drawings are sometimes generated before the completion of the design work by 3D CAD system. The work for creating 3D CAD models is redundant. This redundancy should be eliminated with incomplete design information for the production.

The other problems in production process are delays in the production schedule owing to weather conditions, late delivery of purchases and unexpected problems in resource or facility. These unavoidable troubles which experienced experts manage with their skills in current practice should be handled by software platform to keep the production schedule.

23.4.4 Considerations for Software

As for the deployment of CE, a sophisticated CAD system may allow removal of barriers related to information technology. As shown in this chapter, data sharing across design phases is a key feature for an efficient concurrent design process. In the meantime, shipyards with a long history have a lot of in-house software to facilitate each design process and the data flow from upstream to downstream. This kind of software is useful and shows good performance in local process optimization. The quality of the design may be currently satisfactory for the local process with the use of in-house software. The installation of a single database for all design stages appears to be very difficult and is not yet feasible to handle huge number of parts of a vessel. Single databases and sophisticated design information sharing systems will be a good solution for new emerging shipyards and will be introducing a new concept along with the legacy systems for the shipyards with history. The concept of CE is common and will be deployed with human skills in both types of shipyards by adopting appropriate software systems.

An integrated software system supporting all design phases might be a good solution, but design departments often have their own historical data for in-house legacy software. This kind of in-house software is very useful in case of accessing past design information but may have a risk for maintenance.

23.5 Conclusion

For the deployment of CE in shipbuilding, the following items can be stated in this chapter.

In the early design phase, the trade-offs for many design consideration are to be addressed. Early adoption of 3D models and its utilization for engineering calculation are keys for a design process improvement.

As for basic design, detailed design and production, detailed and well matured design and manufacturing information is required. History data management and simulation techniques for prediction are the key for solving this problem. Organizational changes will also be required to generate design information earlier.

23.6 Future Trend

Accumulation of design cases is huge, while the computational power for detailed simulation is available now. Making decisions for the design and construction process in an early phase might be realistic in the shipbuilding industry.

The shipbuilding process and software for ship design and construction have been described in this chapter. The concept of CE has already been deployed in shipbuilding companies. However, information infrastructures are not enough to support efficient concurrent design and construction processes. The work load in upstream design such as basic design will increase by deploying the CE concept into practice, but is expected to decrease time and costs downstream, by improving the quality of the products and reducing rework. Prediction by simulation and detailed design and construction plans made possible in an early design phase may pave the way to innovation in matured shipbuilding industries.

References

1. Storch RL, Hammon CP, Bunch HM (1988) Ship production. Cornell maritime Pr/tidewater Pub
2. Meijer K, Pruyn J, Klooster J (2009) Early stage planning support. In: Proceedings of international conference on computer application in shipbuilding 2009, vol 2, paper 23
3. Kuutti I, Mizutani N, Kim HS (2011) Efficient integration of 3D design with engineering at the early design stages. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 2, paper 1
4. Fach K, Bertram V, Jefferies H (2009) Advanced simulations for ship design and redesign. In: Proceedings of international conference on computer application in shipbuilding 2009, vol 2, paper 8
5. Papanikolaou A, Zaraphonitis G, Harries S,Wilken M (2011) Integrated design and multiobjective optimization approach to ship design. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 3, paper 4
6. Bons A (2009) QSHIP; Advanced use of hydromechanics in early design stage. In: Proceedings of international conference on computer application in shipbuilding 2009, vol 3, paper 6

7. Ginnis AI, Feurer C, Belibassakis KA, Kaklis PD, Kostas KV, Politis CG (2011) A CATIA(R) Ship-parametric model for isogeometric hull optimization with respect to wave resistance. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 1, paper 2
8. Shibasaki K, Nishimura Y (2009) Utilization of 3D-CAD system at early design stage and powerful interface between 3D-CAD and FE-analysis. In: Proceedings of international conference on computer application in shipbuilding 2009, vol 1, paper 7
9. Nakao Y, Hirai K, Hirayama T, Ito K (2011) High precision basic design. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 2, paper 5
10. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manag 7(2):101–115
11. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manag 7(2):116–131
12. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Syst Manag 6(1):7–24
13. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manag 6(4):307–323
14. de Góngora R (2011) Efficient design of outfitting and machinery spaces. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 3, paper 12
15. Willmes G, Grau M, Pfalzgraf P (2011) Communication processes in shipbuilding based on intelligent 3D PDF documents. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 2, paper 12
16. Hiekata K, Yamato H, Yamato S (2010) Ontology based knowledge extraction for shipyard fabrication workshop reports. Expert Syst Appl 37(11):7380–7386
17. Chung BY, Kim SY, Shin SC, Koo YH (2011) The study on ship compartments arrangement optimization with knowledge-based systems. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 1, paper 3
18. Sun J, Hiekata K, Yamato H, Nakagaki N, Sugawara A (2014) Virtualization and automation of curved shell plates manufacturing plan design process for knowledge elicitation. Int J Agile Syst Manag 7(3–4):282–303
19. Elgh F (2014) Automated engineer-to-order systems. A task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3–4):324–347
20. Sun J, Hiekata K, Yamato H, Nakagaki N, Sugawara A (2014) A knowledge-based approach for facilitating design of curved shell plates' manufacturing plans. In: Cha J et al. (eds.) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering pp 143–152, IOS Press, Amsterdam
21. Larkins D, Roberts P (2011) A process focused approach to ERP integration with CAD. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 1, paper 4
22. Lin ZK, Chiu C, Hsu YB, Shaw HJ (2011) An effective approach for developing an integrated CAD and ERP system in hull. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 1, paper 10
23. Gonzalez C, Alonso F (2011) Advanced CAD-PLM integration in a naval shipbuilding environment. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 2, paper 13
24. Rong L, Soonhung H (2011) Business integration between CAD/CAM and ERP systems in shipbuilding industry. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 2, paper 9
25. Veldhuizen MMJ (2011) Capital project lifecycle management (CPLM) vs. product lifecycle management (PLM) for shipbuilding, marine and offshore industries. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 1, paper 8

26. Park JG, Yi MS, Ha YS, Jang TW (2011) Development of full automatic re-design system for sub-assembly part fabrication of ship blocks. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 1, paper 15
27. Kolich D, Fafandjel N, Storch RL (2011) Lean manufacturing in shipbuilding with monte carlo simulation. In: Proceedings of international conference on computer application in shipbuilding 2011, vol 3, paper 20
28. Storch RL, Lim S, Kwon CM (2011) Impact of customization on delivery schedules. J Ship Prod Des 27(4):186–193
29. Cabos C, Grau M, Wagner L (2013) Product lifecycle management in the shipbuilding and shipping industries. In: 16th international conference on computer applications in shipbuilding (ICCAS), Busan, 24–26 Sep 2013
30. Baumann M, Smidt F, Stevens F, Bauch J, Grau M, Zerbst K (2010) Providing the missing link—integration of siemens PLM NX as CAD system and AVEVA TRIBON M3 as CAM system to a corporate toolset for the design and manufacturing of ship outfitting. In: 4th European conference on production technologies in shipbuilding, Papenburg, 29–30 Apr 2010
31. Pfalzgraf P, Pfouga A, Trautmann T (2012) Cross Enterprise Change and Release Processes based on 3D PDF. In: Stjepandić J (ed) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 753–763
32. Grau M, Liese H, Stjepandić J (2013) Intellectual property protection in the maritime industry. In: 16th international conference on computer applications in shipbuilding (ICCAS), Busan, 24–26 Sep 2013
33. Willmes G, Grau M, Pfalzgraf P (2012) Supply chains in the maritime industry based on 3D PDF documents. In: 5th European conference on production technologies in shipbuilding, Bremerhaven, 26–27 April 2012
34. Wendenburg M (2012) Datenaustausch auf Hochtouren. CADCAM Report, 5/6
35. Hiekata K, Yamato H, Enomoto M, Kimura S (2011) Accuracy evaluation system for shipbuilding blocks using design data and point cloud data. In: Frey DD et al (eds) Improving complex systems today. Proceedings of the 18th ISPE international conference on concurrent engineering. Springer, London, pp 377–384
36. Hiekata K, Yamato H, Enomoto M, Oida Y, Furukawa Y, Makino Y, Sugihiro T (2010) Development of accuracy evaluation system of curved shell plate by laser scanner. In: Pokojski J et al (eds) New world situation: new directions in concurrent engineering. Proceedings of the 17th ISPE international conference on concurrent engineering. Springer, London, pp 47–54
37. Klyne G, Carrol J (2004) Resource description framework (RDF): concept and abstract syntax. W3C Recommendation. http://www.w3.org/TR/rdf-concepts/. Accessed 15 Nov 2013
38. Mitsuyuki T, Hiekata K, Yamato H, Haijima K (2013) A study on evaluation of organizational performance considering the workers and facilities. In: Stjepandić J et al (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 533–544
39. Hiekata K, Grau M, Stjepandić J (2014) Case studies for concurrent engineering concept in shipbuilding industry. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy, pp 102–111. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, 2014

Chapter 24
Consumer Goods

Chun-Hsien Chen, Li Pheng Khoo and Nai-Feng Chen

Abstract Product design and development (PDD) has shifted its focus from addressing functional and technological issues to user-centric and consumer-oriented concerns in recent years. More specifically, the experiential aspect of design has taken a crucial role in creating more consumer-focused products. Often, customer research or user-involvement studies are conducted to explore necessary knowledge and gain an insight into user experience. Unlike functional requirements, experiential customer requirements are usually more tacit, latent and complex. As such, the issues concerning user experience exploration in consumer goods design deserve more attention. These will be the focus of this chapter. In this regard, a prototype context-based multi-sensory experience system (CMSES) with a scenario co-build strategy (SCS) is proposed to facilitate user experience exploration in designing consumer goods. A three-stage case study is employed to illustrate the proposed prototype system. Potential of the proposed approach in the context of concurrent engineering (CE) and collaborative product development (CPD) is discussed.

Keywords Customer requirement management · Multisensory experience · Product design · User experience · User involvement

C.-H. Chen (✉) · L.P. Khoo · N.-F. Chen
School of Mechanical and Aerospace Engineering, Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798, Singapore
e-mail: mchchen@ntu.edu.sg

L.P. Khoo
e-mail: mlpkhoo@ntu.edu.sg

N.-F. Chen
e-mail: nfchen1@e.ntu.edu.sg

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_24

24.1 Introduction

The philosophy of concurrent engineering (CE), and its successor collaborative product development (CPD), has been widely applied in product design and development (PDD) for decades. It addresses issues caused by the sequential development process, which is usually lack of communication between different functions of a company and requires long development times [1, 2]. On top of CE and CPD, the evolution of the PDD paradigm goes on because consumer goods are becoming more and more complex and customers generally expect more variety, lower costs, better performance, higher quality and more rapid advancement [3]. By properly incorporating the latest development in the realm of CE and CPD into the PDD process, companies may gain a competitive edge and have better opportunities to produce higher quality and cost-effective consumer goods in shorter time [4]. The key concept of CE and CPD is the early consideration and involvement of all relevant elements of the product life cycle (PLC) [5]. Accordingly, cooperation between multidisciplinary teams is indispensable. It has become a must to simultaneously consider much more complex requirements from different stakeholders by these teams [6].

As an implementation of CE and CPD, this chapter deals with the issues concerning user experience exploration in consumer goods design. More specifically, a prototype context-based multi-sensory experience system (CMSES) with a scenario co-build strategy (SCS) is proposed to facilitate user experience exploration. To illustrate the CMSES, this chapter starts with introducing the current trend of PDD in Sect. 24.2. This is followed by a description of the proposed methodologies, i.e. the CMSES and SCS, in Sect. 24.3. Subsequently, a three-stage case study on a biscuit container design is used to demonstrate the CMSES and SCS in Sect. 24.4. After that, Sect. 24.5 gives a general discussion regarding the case study and highlights the potential of applying the 'context-based multi-sensory experience exploration and design'. The last section, Sect. 24.6, summarizes the main conclusions reached in this chapter.

24.2 Related Work

Owing to the paradigm shift of the PDD process in recent years, apart from addressing functional and technological issues, user-centric [7, 8] and consumer-oriented [9] concerns have proven themselves to be as equally, if not more, important in developing a successful product. As a result, fulfilment of customers' needs and wants has become inevitable. Therefore, it is important to treat users or customers as stakeholders and invite them to contribute their views in the 'fuzzy front-end' of product development. The early user/customer input, knowledge integration and decision making may have a crucial influence on the cost, time-to-market, and the success or failure of a product, especially in the context of new product development

(NPD) [10–12]. In this regard, companies often conduct user involvement studies to discover and identify the genuine voice of customers (VOC) [13].

Moreover, in order to create more consumer-focused and successful products for the emerging experience economy that emphasizes selling experience [14, 15], companies should further concentrate their endeavours on the experiential aspect rather than merely the material one [16–19]. In other words, the VOC should include explicit, tacit, tangible and intangible customer requirements and the effort should be extended to an experiential level. To realize this idea in NPD, researchers encourage product developers and designers to treat users as experts of their own experience, explore potential user experiences and design for experience [20–22]. By exploring knowledge regarding user experience before and during design conceptualization, a company can better plan for its marketing and design strategies at an early stage and can be more confident about the product and its experience created to gratify the users.

However, during user experience exploration, designers may face some difficulties due to the fact that user experience is inherently complex, subjective and dynamic [23]. The characteristics and corresponding aspects of user experience are organized as shown in Fig. 24.1. Some issues might rise readily if such inherent nature of experience has not been taken into careful considerations when exploring it. Three major aspects of issues are identified as follows.

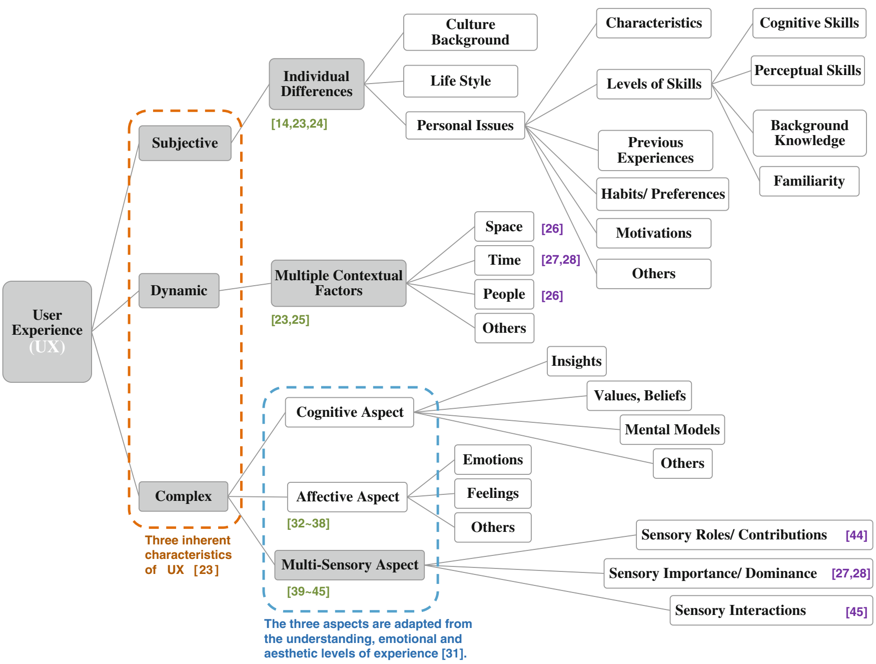


Fig. 24.1 Different aspects of user experience

First (see Fig. 24.2), as an experience is inherently personal and exists only in the mind of an individual [14], it is important to treat users in a more personal manner [24]. Without considering the subjective nature, a company may treat users in a too general manner and ignore crucial individual differences. Consequently, it may lose some valuable customer segments in a highly competitive business environment.

Second (see Fig. 24.3), as a user's experience is dynamic and context-dependent [23, 25], it is inevitable to consider how multiple contextual factors may influence user experience. For example, 'companionship' is a powerful factor to have an impact on user experience [26]. It also requires attention that a user's multi-sensory experience may vary dynamically at different usage phases [27, 28]. Without tackling the contextual factors, user experience or evaluation may become ecologically invalid [29]. In addition, it is better to avoid treating a product as a starting

Fig. 24.2 User experience is subjective due to individual differences

Fig. 24.3 User experience is dynamic due to multiple contextual factors

point. Instead, designers are encouraged to redefine ‘a product’ as ‘a context for experience’ [30] and develop ideas from the ‘contextual level’, through the ‘user-product interaction level’, then to the ‘product level’ [22]. In doing so, designers can have a better exploration of user experience and prevent to become stuck by current designs especially for NPD.

Third (see Fig. 24.4), owing to its diverse and complex nature, user experience is difficult for designers to explore and discuss in a more comprehensive way. For example, it may cover cognitive, affective and sensory aspects as suggested by Hekkert [31] in his work on the three levels of experience: understanding, emotional and aesthetic. Recently, more and more researchers advocate the great value of emotional design [32, 33], affective design [34, 35] and Kansei (*a Japanese word for sensory*) engineering [36–38].

Furthermore, multi-sensory experience design has also attracted more and more attention since user experience is closely related to how the senses are stimulated and gratified [39–41]. In addition, it should not merely focus on visual aesthetics but should consider all of the senses [42]. Actually, user experience can be enriched to a certain extent if there are more sensory modalities involved [40] and more sensory memories activated [43]. Researchers have studied sensory experience from several facets such as roles of the senses [44], sensory importance [27, 28] and various kinds of interactions between senses [45].

Especially in the highly competitive era, companies are tackling much more complex design problems which no longer involve merely functional or cognitive aspect. It is quite a challenge for designers to deal with information or data of different format and characteristics, especially when experiential, intangible and

Fig. 24.4 User experience is complex and comprises many different facets

tacit elements are involved. Thus, the crucial key to create successful products lies in the integration of multiple factors from experiential, contextual and sensory aspects starting from the early stage of PDD [26].

In a nutshell, the main challenge is to concurrently deal with all these inherent characteristics of user experience during user involvement studies. Nevertheless, current studies seldom take these essential characteristics into more careful considerations during user-experience exploration. In addition, more practical studies, which demonstrate how designers can explore and discuss users' multi-sensory experience in a more in-depth and comprehensive manner, are still lacking. Based on these understandings, this study investigates the 'context-based multi-sensory experience exploration and design' to help designers get an in-depth understanding about user experience so as to facilitate experience design.

Although it is no easy task for a company to control or predict experience needs accurately [46], it is possible and justifiable to provide customers with their desired experience based on some prerequisites [46, 47] or exclude some potential negative experience. The deeper and more comprehensive the designers can understand user experience, the higher the possibility for them to create long-lasting pleasing products. Therefore, it is worthwhile to develop methods and tools to strengthen user-experience exploration.

24.3 Methodologies

To facilitate the effectiveness and efficiency of 'context-based multi-sensory experience exploration and design', a prototype CMSES with a SCS is proposed. The CMSES and SCS are demonstrated using a three-stage case study on a biscuit container design. Details are presented in the following sub-sections.

24.3.1 Context-Based Multi-sensory Experience System (CMSES)

A prototype CMSES is established to tackle the issues mentioned in the previous section. The system attempts to guide the PDD process from user-involvement studies to design conceptualization in a user-centric, consumer-oriented and experience-focused manner. To address the subjective, dynamic and complex nature of an experience, CMSES possesses the following characteristics.

1. User experience is explored under a specific usage context, which can prevent ecologically invalid results.
2. Individual differences are taken into consideration when tackling multiple contextual factors, such that the result can be more reliable and closer to a user's real situation.

3. As humans first perceive stimuli from their senses, it handles the multiple aspects of user experience starting from the sensory aspect then progressively bringing in more and more aspects.

Corresponding to the first and second characteristics, i.e., the subjective and dynamic nature of an experience, a SCS is applied. As shown in Fig. 24.5: Ⓐ: the SCS invites users in the decision making of scenario building, which was usually done by design teams and users can merely play a passive role. With the SCS, designers and users together may co-build a more customized scenario which can link up with one's real life more closely [26]; Ⓑ and Ⓒ: Customized scenarios can not only help to strengthen user-experience exploration at a later stage of user involvement studies but also provide valuable feedback of individual differences regarding usage context to the marketing department; Ⓓ: In this case, users may experience a product like they normally do in real life and may possess a feeling of ownership. By treating a product as one's own property, a user can become more 'emotionally attached' and be motivated to share more of his/her experience [48]. In doing so, the user-involvement process can be more relaxing, inspiring and creative; Ⓔ: Consequently, designers can explore more reliable and valuable feedback of user experience and evaluation. Corresponding to the third characteristic, i.e., the complex nature of an experience, designers can examine user experience in a more

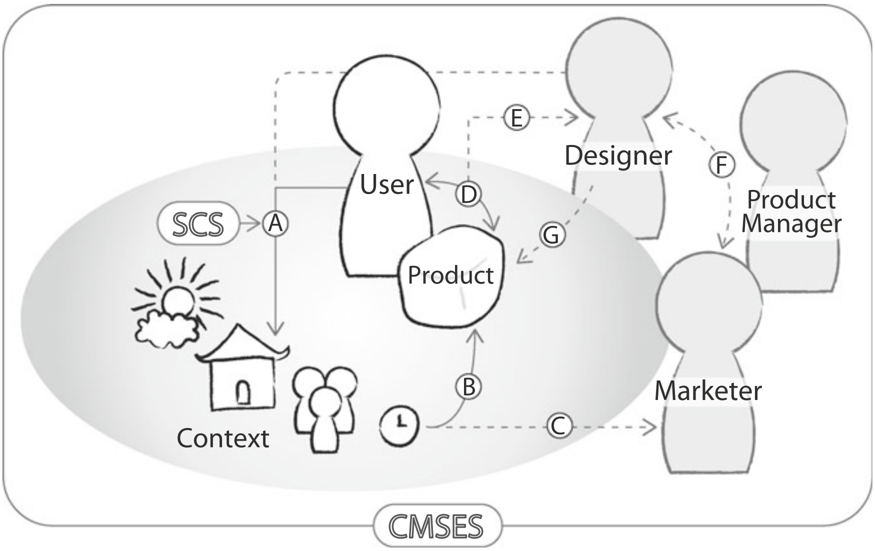


Fig. 24.5 The context-based multi-sensory experience system (CMSES). *Note* Ⓐ: Scenario co-building (based on a user's real situation). Ⓑ: Customized scenario(s) (for the user to experience the product); Ⓒ: Feedback of individual differences regarding usage context; Ⓓ: Context-based user-product interaction (user experience); Ⓔ: Feedback of (multi-sensory) user experience and evaluation; Ⓕ: Design and marketing strategies concerning user experience; Ⓖ: Context-based multi-sensory experience design

detailed and comprehensive way by connecting the multiple sensory aspects with the cognitive and affective aspects. The general goals are to get an in-depth understanding of user experience and identify opportunities for multi-sensory experience design; Ⓕ: Accordingly, a company can better plan for design and marketing strategies concerning user experience; Ⓖ: As a result, designers can conduct multi-sensory experience design from a contextual level in a more manageable manner so as to create a more long-lasting positive product experience.

24.3.2 Scenario Co-build Strategy (SCS)

Scenarios, describing the usage context, are frequently applied in a user involvement study for users to evaluate a system or product [49]. However, conventional scenarios are usually set up by design teams and, hence, fail to systematically take individual differences into more careful consideration. Thus, it may hardly reflect users' real life because of strong individual differences (e.g., different cultural background, life style, personal habits) in numerously varied societies worldwide [26]. The problem is crucial since user experience can be affected by multiple contextual factors defined in a scenario. In order to explore more useful and valuable user experience, the scenarios used in a user involvement study should fit into user's real life as much as possible [26]. In addition, users should not play a passive role during user participation. Instead, users should be treated as 'experts of their own experience' [22] and own the freedom to build their own experiences [17]. Based on such understandings, a SCS is established to address the issue by providing users with the opportunity and freedom to decide the scenario that is the most suitable for them [26]. The idea is for users and designers to co-build more customized scenarios (Fig. 24.6) so as to strengthen user experience exploration in user involvement studies. The co-built scenarios may have different structures and levels of freedom, depending on how much and how detailed users can contribute to the context settings of a scenario.

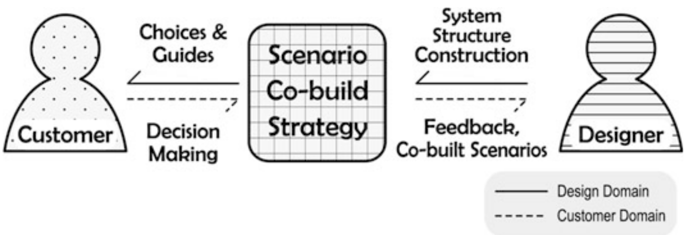


Fig. 24.6 The scenario co-build strategy (SCS); (adopted from Chen et al. [26])

24.4 Case Studies

To demonstrate the CMSES and SCS, a three-stage case study was conducted. At the first stage, users' multi-sensory experience and evaluation were explored and discussed. At the second stage, five biscuit container design concepts were generated based on the experiential knowledge gained. Subsequently, a survey was carried out to evaluate the concepts, and then a preferred concept is chosen and improved at the third stage.

24.4.1 The First Stage

The main purpose of the first stage of the case study is to explore knowledge regarding users' multi-sensory product experience from a contextual point of view. Based on the CMSES and SCS, this study illustrates how designers can explore and discuss users' multi-sensory experience concerning multiple contextual factors and diverse individual differences.

24.4.1.1 Methodology

The product chosen in the case study is 'Mary Biscuit' of the design brand 'Alessi' (see Fig. 24.7). The design features, e.g., the biscuit shaped lid with vanilla scented and special texture, make 'Mary Biscuit' stand out from other competitors. Five participants were invited to perform a pilot study and 33 participants (Mean age = 23.36 years, age range: 20–26 years; 15 female and 18 male) were invited to perform the formal experiment.

Based on the scheme of SCS proposed by Chen et al. [26] (see Fig. 24.8), this study investigates two main phases of user experience, namely the trial and usage phases. At the trial phase, the users purchase the product and interact with it for the first time. At the usage phase, the users get to know more about the product and have fresh experience with it.

Fig. 24.7 'Mary Biscuit' of the design brand 'Alessi'

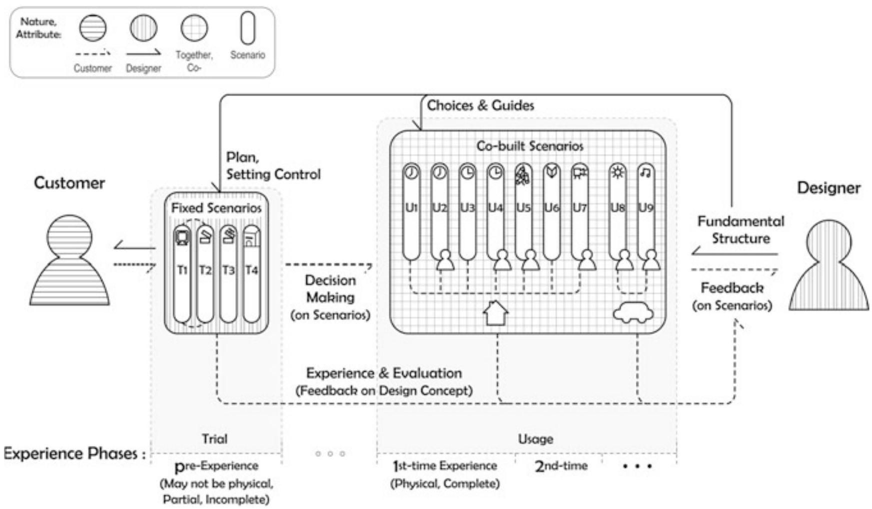


Fig. 24.8 One possible scheme of SCS; (adapted from Chen et al. [26])

More specifically, each participant experienced and evaluated 'Mary Biscuit' under both trial scenario(s) and usage scenario(s). At the beginning, trial scenario(s) were assigned to participants for them to experience 'Mary Biscuit' for the first time. Subsequently, based on the experiences and understanding about the product, participants can then choose preferred usage scenarios at the usage phase. The trial scenarios include three basic situations of the obtainment of the product while the nine usage scenarios cover both daily use and special events (see Table 24.1).

The experiment was conducted in a laboratory. There were two laptops to present scenarios and questionnaires respectively and two video cameras to record the process. The time was controlled within 45–60 min. Microsoft PowerPoint (PPT) slides were used to guide participants choosing preferred scenarios and representing the chosen ones. Scenarios are represented through descriptors (textual narratives), images (of the context) and videos (only for online webpage scenario) to help participants build a mental model of the usage occasions. Both qualitative methods (i.e., think-aloud protocol (TAP), observations and interviews) and quantitative questionnaires were used to collect different kinds of information and data.

24.4.1.2 Results and Discussions

To better represent the knowledge regarding users' multi-sensory experience, the qualitative and quantitative results were put together and the discussion was separated according to seven issues, A–G, as follows.

A. **Sensory importance**
   To discuss the multi-sensory experience, sensory importance is first examined. Participants' attitudes toward the importance of each sense were solicited using

Table 24.1 Scenarios and participants record

Phase (way to provide scenarios)	Code	Scenario	Note	No. of participants	Participant code
Trial (assigned)	T1	Virtual shopping 1: online webpage	Buy, virtual	11	4, 12, 20, 21, 22, 23, 26, 28, 29, 30, 31
	T2	Virtual shopping 2: receiving order			
	T3	Receiving gifts at home	Granted, physical	11	1, 2, 3, 5, 6, 7, 10, 15, 19, 24, 32
	T4	Physical shopping at store	Buy, physical	11	8, 9, 11, 13, 14, 16, 17, 18, 25, 27, 33
Usage (self-choosing)	U1	Breakfast, alone	Indoor, daily use	2	1, 7
	U2	Breakfast, with family	Indoor, daily use	1	15
	U3	Afternoon tea, alone	Indoor, daily use	1	11
	U4	Afternoon tea, with family	Indoor, daily use	6	8, 22, 24, 28, 30, 31
	U5	Festive seasons, (e.g., Chinese new year), with companion	Indoor, special event	11	3, 4, 5, 6, 10, 14, 18, 21, 25, 26, 32
	U6	Examinations, alone	Indoor, special event	6	2, 9, 12, 19, 20, 29
	U7	Movie marathons, with companion	Indoor, special event	3	13, 23, 27
	U8	Picnic, with Companion	Outdoor, special event	9	10, 16, 17, 18, 24, 27, 28, 30, 33
	U9	Attending a party, with companion	Outdoor, special event	1	12

questionnaires after each scenario experiencing. The overall results show that vision is the most important sense, followed by touch, olfaction and audition. Though for some scenarios the average scores seem to be higher, there is no significant difference.

In the afterward questionnaire, participants' attitudes towards what kinds of design features can affect their purchasing decisions were further solicited for those who experienced T1, T2 and T4 scenarios (in which the obtainment of the product is buying but not granted as a gift; there were totally 22 participants). The results suggest that most participants care about tactile experience (counted 19 times), followed by special features (which can relate to any sensory experience once it shows the uniqueness compared to others) (18 times), and the visual experience (17 times). Seven participants thought that olfactory experience is also a concern while no participant gave credit to auditory experience.

It is known that there is usually a dominant sense during user-product interaction that can collect more information, have more influence, and attract more attention [39]. Yet, the sensory dominance or importance may be affected by product types [50], product characteristics [44] and the stages of usage [27, 28]. Although the study included both trial and usage phases that participants experienced 'Mary Biscuit' for at least two times, the experiences were still very fresh and belonged to early phases of product usage (e.g., the early stage of product experience [27] or the buying stage [28]). Usually at such phases, vision can have more influence and importance than other senses [27, 28]. Furthermore, under most situations vision can gather most information of a product in the shortest time compared to other senses [39, 44]. All these could be the reasons why sensory dominance did not shift significantly and vision was the dominant sense at both trial and usage phases in this study.

B. **Visual experience**

Vision, in this case, plays an important role in both cognitive and affective aspects of user experience. First, participants relied heavily on vision to form the first impression, explore the functions, examine and evaluate the product. Especially in the scenarios in which the main function of the product is unrevealed (as in T4 scenario participants find the product on the showcase along with other goods without being informed it is a cookie container), participants mainly inferred the product function from the overall form (as a container) and the biscuit shaped lid (as to store cookies). This result is in line with Alessi's design philosophy 'form follows function'. Second, the product's overall form can elicit many kinds of associations including pillow, cushion, wrist rest, dog bone, UFO, red blood cell, fish tank, tissue box, flowerpot, massager, lamp, chair and stool. Such associations can further influence participants' behaviour. For example, some participants really treated 'Mary Biscuit' as a pillow to lay it near the head or a cushion to hug in the arms. Third, a lot of interactions were aroused by the biscuit shaped lid (along with other sensory design features: colour, texture, scent), e.g., "the colour and texture of the lid, quite realistic smell, and feels like real biscuit makes me want to eat biscuit (P26, Interview, T2)" and "[pranking friends] hey, there's a biscuit, do you want to eat? (P10, Interview, U5)". Fourth, plenty positive emotions (e.g., happiness, surprise, funny, interesting and satisfaction) can be evoked by the pleasing form of the container as well as the biscuit shaped lid while some negative emotions (e.g., disappointment, worry and concern) can be evoked by the middle-line (i.e., a join line between the lower and upper parts as shown in Fig. 24.7) of the container. Yet this middle-line of the translucent container is related to not only the aesthetic aspect but also the functional aspect of the product such as durability; e.g., "makes me worry that the Mary will break easily (P26, Interview, T2)".

C. **Tactile experience**

It is found that the form of the product can further affect participants' tactile experience. For instance, the middle-line of the container may become a disturbance while touching the overall form, which consequently affects participants' feelings, e.g., "I don't like this [act: touch the middle-line]. I'd like it to be one piece (P28, Interview, T2)". Take the concave shape of the bottom as a positive example, many participants praised that the shape is friendly for hands to hold, e.g., "if I want to hold it by one hand to serve people, it's quite easy to hold (P6, Interview, U5)". There is also the case that through touching, participants can explore the more detailed form apart from merely viewing the product, especially when there are some special shapes or irregular contours, e.g., "you have to feel and touch the shape, you can't really visually see the shape, like the bottom, you can't see the specialty of it until you touch it (P15, TAP, T3)".

Besides the form, the special texture also matters a lot, because it can bring more positive evaluation as well as evoke more positive emotions. In fact, it is suggested that designers should take very good care of every possible tactile feeling that a product may bring to users. On the one hand, tactile experience can affect most participants' purchasing decision as the results shown above. On the other hand, participants would examine the product in a more detailed way through touching, thus, even minor matters can become a plus or minus point. For instance, the edge of the opening of the container can be an issue, e.g., "I like the edge! Because it's not like normal containers that have quite a few jagged lines, this is really smooth (P29, TAP, T2)".

D. **Olfactory experience**

In terms of olfactory experience, participants showed great individual differences from various aspects. First, participants reported diverse individual sensory preference or habits toward the vanilla-scented lid. On the one hand, some participants held a quite positive attitude toward the scent and they may highlight this feature to their (imagined) companion, e.g., "I tell you what, this box is really special eh ~ come smell smell smell (P5, TAP, U5)". On the other hand, some participants kept a negative attitude and some even stated that they may choose not to buy it, e.g., "I am not a vanilla person, if it's too strong, I won't buy it (P26, Interview, T1)". Nevertheless, there were also few participants who did not consider the scent as an issue. In fact, participants' personal preference is closely linked with affective emotions. For participants who liked the smell, they tended to smell the lid for more than once and some positive emotions also came along, e.g., "like to go back and smell over, over, and over again (P23, TAP, T2)" and "because of the smell, very happy, since I bought the right thing (P4, TAP, T2)". Participants who did not like the smell would also behave greatly with strong emotions. Additionally, the emotions evoked by the scented lid (either positive or negative) are much stronger than those evoked by other stimuli. It is known that olfaction has a very strong connection with emotions and relates closely to ones' personal experiences as well as memories [51]. As highlighted by Spence [51], the importance of olfaction is obvious and "the products of tomorrow will embrace the olfactory revolution (p. 3)." Indeed, 'Mary Biscuit', with such olfactory feature, is found to be able to bring users quite vivid emotions and differentiate itself from other containers. Furthermore, the sensory preference can further affect the way how participants treated the lid and their attitude with regard to the fading away of the scent after 1–1.5 year. Generally, those who adored the scent may choose not to wash the lid and may wish the scent could stay longer. Whereas those who dislike the scent claimed that they wish to wash the scent away and may like 'Mary Biscuit' more once the scent faded away.

Second, there is diverse individual difference in the interpretation of possible interactions between the vanilla-scented lid and the food stored inside. For example, a few participants looked on the bright side and imagined the taste of the cookies stored inside may become tastier. Yet some participants not only worried the food inside may be affected by the scent in a negative way but also were concerned about the artificial smell in a food container could be harmful to the body. As a result, the scent may affect participants' decision-making regarding what to contain. Some participants claimed that they would not use it to contain food, some would choose biscuit with wrappers so that the flavour (of biscuit) will not be affected, some may avoid strong scented food so that the scent (of the lid) will not be affected and some stated they would choose the biscuit that can match the flavour of scent.

Despite the fact that there are great individual differences in participants' sensory habits and preferences, the special olfactory feature does help 'Mary Biscuit' stand out from other competitors; not only because the scent can evoke plenty emotions and enrich one's affective experience but also because it can be an icebreaker to open a topic and enable many interactions among people. However, while designing this kind of special sensory features, designers should be careful about possible sensory interactions. As Schifferstein and Desmet [52] addressed, a product's final success depends on how "all senses" are stimulated and gratified. Therefore, it is important to design a more "natural, logical and coherent" sensory experience. In this case study, the colour scheme of 'Mary Biscuit' chosen is 'ice', in which the container is translucent and the lid has normal biscuit colour. It seems to be logical and coherent when the biscuit colour matches the vanilla scent. Nevertheless, other colour schemes of 'Mary

Biscuit' provided by Alessi also include white, orange, blue and green; yet there is only one flavour of smell—the vanilla scent. In the study, participants can observe the different colour schemes from the online shopping webpage (in T1), from the package while receiving as a gift (in T3) or by asking the clerk to show them (in T4). One participant originally said he preferred green colour. However, after he knew the scent is still vanilla flavour he changed his mind immediately. Furthermore, he commented the combination is "definitely a wrong design concept (P8, Interview, T4)". In addition, he gave some suggestions such as "not only depending on colour preference, but also smell preference we can choose (P8, Interview, T4)". Besides the sensory interactions, it is also worthwhile for designers to identify how the context may affect one's sensory preference. For example, one participant noted while experiencing the 'Examinations (U6)' scenario that "if I'm studying, I'd rather this [the lid] is coffee scented (P29, TAP, U6)".

Considering possible individual sensory preference, the afterward questionnaire solicited participants' preferences toward the flavour of the scent. The results show that 17 participants preferred no smell at all while 12 participants chose the same vanilla scent, 11 for fruits scent, 3 for other cookies scent and no participant considered perfume style. Thus, while planning or designing a product and its multi-sensory experience, design teams should have more careful consideration including possible sensory interactions as well as diverse individual sensory preference and habits.

E. **Auditory experience**

In discussing auditory experience, the product sound considered here is the sound produced while closing the lid, which is found closely related with the functional aspect—the tightness of lid. This can be an important issue especially for such product as a food container where 'air-tight' is one of the essential considerations. It is commonly believed by participants that if the lid can close tightly, there should be a clear clicking sound as a feedback while pressing down. Hence, when a light and soft sound replaced a clear clicking sound, many participants showed their concerns. As a result, some negative emotions (e.g., worry, concern, disappointment and unsafe) were evoked due to the problem. Similar to other senses, there are individual differences in personal sensory preference and the interpretation of the product sound. In the case study, a few participants noted the positive aspects of the light and soft sound, e.g., "it is well designed and well-constructed (P28, Interview, T2)" and "normally there must be a loud sound to close it tightly, but this can close tightly yet the sound is quiet (P33, Interview, T4)". Besides, some participants found that since 'Mary Biscuit' is less noisy than other containers, it can fit into some special usage occasions well, e.g., "snack in middle of night will feel less guilty (P29, Interview, T2)" and "Mary is suitable for the library since it got no sound (P29, Interview, U6)".

F. **Sensory design features and usage occasions**

Based on the results of the case study, it is suggested that sensory design features might be able to affect a product's suitable usage occasions. In this sense, besides designing from a contextual level [22], it is further suggested that designers should always return to the contextual level after design conceptualization to check whether different sensory design features could help the product fit well to the targeted usage contexts or not.

G. **Individual sensory preference and designs in the market**
Among all the multi-sensory experiences, it seems there are more individual differences in the olfactory and auditory experiences. The differences may lie in several aspects including personal sensory preference (e.g., like/dislike a sensory design feature), personal sensory habits (e.g., care how much about a sensory gratification or have special concerns regarding a sensory experience), individual interpretation of a stimulus or phenomenon (e.g., view an event from different points of view) and personal sensory sensitivity (e.g., initial feelings of perceptions or physical sensitivity toward a stimulus). Nevertheless, most products in the market today still primarily emphasize the visual aesthetic [28], some may further secondarily strengthen the tactile gratification, but few would consider special olfactory or auditory experience unless the product is directly or strongly related to smell (e.g., perfume and deodorant) or sound (e.g., musical instruments and washing machine). In other words, while purchasing a product, usually a customer can have plenty of choices of various forms, sizes, colours, materials and even textures but may have little or no choice of special olfactory or auditory features (except certain types of product, e.g., perfume). Based on the example of olfactory experience illustrated, participants had diverse needs and wants regarding the olfactory experience yet there was no opportunity for them to choose their preferred sensory features, e.g., the intensity or flavour of the scented lid. As a result, not only some segments of customers cannot be satisfied but also the company may lose some segments that tend to have stronger personal sensory preference or sensory habits.
Hence, it can be a good opportunity for a company to design for all senses (i.e., multi-sensory design) in a more considerate way contemplating various individual differences to satisfy more segments and differentiate a product from many competitors. For example, the concept of 'mass customization' can be applied to increase sensory design features (and sensory experiences) variety while controlling manufacturing cost in order to create more customized and personalized product that can fit individual customer's condition better.

24.4.2 The Second Stage

The main purpose of the second stage of the case study is to carry out concept generation for the biscuit container design. Based on the experiential knowledge gained from the first stage, 15 design concepts were created for the young segment. These concepts were further categorised into five groups and, subsequently, a final design concept was chosen and revised from each group. Thus, five biscuit container design concepts in total were generated. In order to conduct a customer survey at the later stage, CAD (Computer Aided Design) models of all 5 designs as well as 'Mary Biscuit' were created using SolidWorks software.

The first design, 'Mushy' (see Fig. 24.9), is a marshmallow-shaped biscuit container with matte surface. The internal surface of Mushy is in brown colour, which suggests the container is filled with chocolate and, hence, may increase users' appetite. There are two ants heading towards the overflow chocolate. The white colour external surface of the 'Mushy' causes this part to become the highlight of the design. The intention is to make users feel curious and make them investigate what is inside the 'Mushy'. The snap-fit lid is easy to open. Besides, there is sweet scent on the external surface of the lid. Users can choose the scent they desire while purchasing 'Mushy'. Available sweet scents include caramel, chocolate, vanilla and honey.

The second design, 'Passion' (see Fig. 24.10), is an orange-shaped biscuit container with matte surface. The matte surface is to suggest similarity of this biscuit container with real orange. This container comprises a tissue paper chamber and six compartments for users to contain and sort various cookies. The transparent biscuit container lid allows users to see the cookies inside the container without opening it. As the white arrow shows, users can rotate the compartment around the tissue paper chamber to search for cookies they desire. Each of the individual compartments can be taken out easily by simply lifting it upward. Both biscuit container lid and tissue paper chamber lid are snap-fit. The tissue paper chamber lid is further customized with fruit scent.

The third design, 'Desire' (see Fig. 24.11), is a snap-fit container with chocolate shape and is made of brown matte plastics. There is a stickman, which is a tissue paper container, lain on the top of 'Desire'. The happy and bright smile on the stickman may cheer users with some positive emotions. Users can play with the stickman and have more interactions with their companion. The tissue paper

Fig. 24.9 The first design: 'Mushy'

Fig. 24.10 The second design: 'Passion'

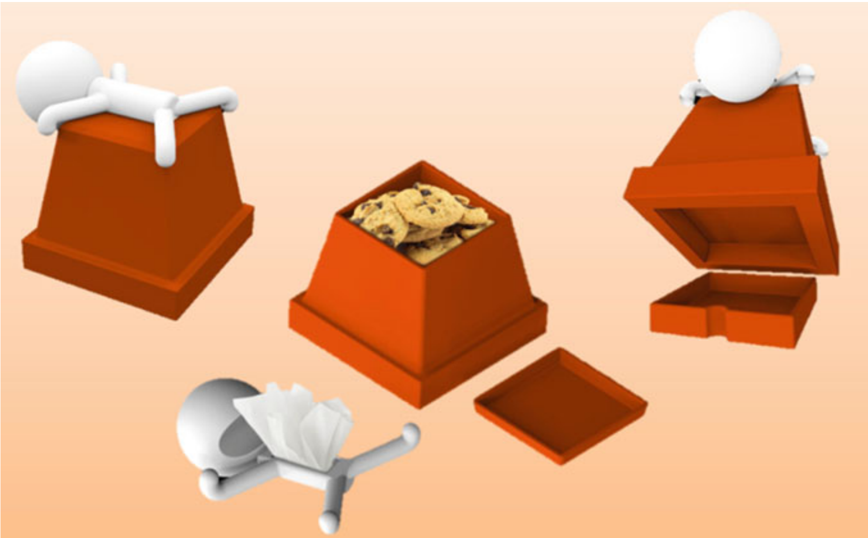


Fig. 24.11 The third design: 'Desire'

container has two choices of scents, which are chocolate and milk. A temporary waste storage, hidden at the bottom of the container, is designed for users to throw their waste easily and conveniently.

The fourth design, ‘Sharkie’ (see Fig. 24.12), is a shark-shaped biscuit container with white colour and glossy surface. It is a screw lock container. There are some functions available. Users can estimate the level of cookies inside ‘Sharkie’ through the transparent eyes. ‘Sharkie’s mouth is a tissue paper holder. There is an inbuilt cutter in the fin for users to open food packaging. As shown in Fig. 24.12, users can slide the food packaging downward through the inbuilt cutter to create a small opening in the packaging. The cutter is built in the fin to ensure safety of users especially children. The slit is designed narrow enough to avoid children to put in their fingers. The fin comes with a variety of scents for users to choose, including fruits, sweet, mint, coffee, perfume, etc.

The fifth design, ‘FreshMint’ (see Fig. 24.13), is a tooth-shaped biscuit container with glossy surface. It is a snap-fit container. There is a toothpaste-shaped and mint-scented tissue paper container on the top of the design. The intention is to educate users. When users clean their lips with tissue paper after eating, the mint smell reminds them of tooth paste. This may further remind users to brush their teeth after eating.

In order to compare the design concepts generated with ‘Mary Biscuit’, the CAD model of it was also created (see Fig. 24.14). The representative sensory design features of the 6 biscuit containers were summarized as shown in Table 24.2.

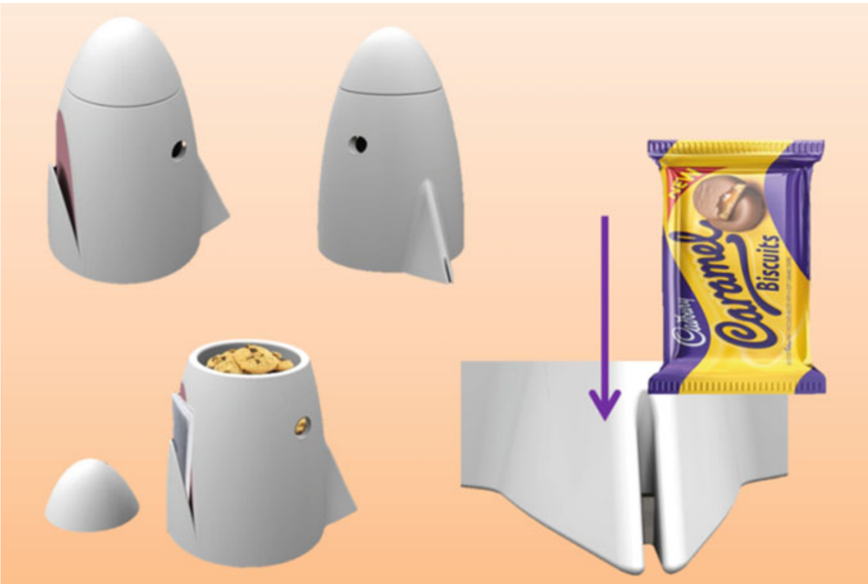


Fig. 24.12 The fourth design: ‘Sharkie’

Fig. 24.13 The fifth design: 'FreshMint'

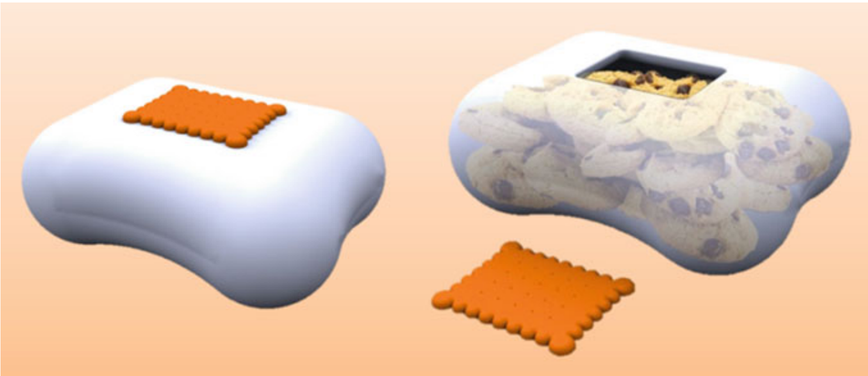


Fig. 24.14 The CAD model of 'Mary Biscuit'

24.4.3 The Third Stage

24.4.3.1 Methodology

The main purpose of the third stage of the case study is to investigate the potential user experience and generate a more consumer-focused design. Therefore, a customer survey was conducted to evaluate the 6 design concepts, namely 'Mushy', 'Passion', 'Desire', 'Sharkie', 'FreshMint' and 'Mary Biscuit', and explore potential user experience. Four (4) participants were invited to go through a pilot study and 33 participants (Mean age = 23.375 years, age range: 19–26 years; 17 female and 16 male) were invited to accomplish the survey. They were recommended to complete

Table 24.2 The main sensory design features

Design						
Sensory design features	Mushy	Passion	Desire	Sharkie	FreshMint	Mary biscuit
Visual	Marshmallow shape, matte surface, ants, overflow chocolate, brown colour internal surface, snap-fit lid	Orange shape, matte surface, tissue paper chamber, compartments, transparent side of compartment, snap-fit lid	Chocolate shape, matte surface, stickman-shaped tissue paper container, temporary waste storage, snap-fit lid	Shark shape, glossy surface, transparent eyes, inbuilt cutter, screw lock lid	Tooth shape, glossy surface, toothpaste-shaped tissue paper container, snap-fit lid	Pillow-shaped container, biscuit-shaped lid, matte icy surface, translucent container, snap-fit lid
Tactile	Matte surface, snap-fit lid	Matte surface, snap-fit lid	Matte surface, snap-fit lid	Glossy surface, screw lock lid	Glossy surface, snap-fit lid	Matte surface, snap-fit lid
Olfactory	Sweet scent (selectable)	Fruit scent (selectable)	Scent (selectable)	Scent (selectable)	Mint scent (selectable)	Vanilla Scent
Auditory	Snap-fit lid	Snap-fit lid	Snap-fit lid	Screw lock lid	Snap-fit lid	Snap-fit lid

the survey with their first thought answer. To avoid bias, no participants at the third stage were involved in the first stage of the case study.

The survey was presented using PPT slides. It consisted of two major parts, viz. individual evaluation and comparison evaluation. In individual evaluation, participants evaluated the design concepts one by one and they were told not to compare the six biscuit containers. The sequence of the biscuit containers appearing in the survey is randomised to reduce bias. The first part of the survey, i.e., individual evaluation, covered the following six aspects.

1. Introduction of the main concept and the design features of the biscuit container design
2. Evaluation on the appearance, functional experience and emotional experience using a 5-point scale
3. Decision making on the preferred scents (olfactory design features)
4. Decision making on the preferred usage occasions
5. Selection of attractive (sensory) design features (listed in Table 24.2) and providing reasons
6. Extra comments and feedbacks.

During the comparison evaluation, participants compared all six biscuit containers and ranked them based on their functional experience, emotional experience and willingness to buy. In addition, participants' attitudes toward the importance of different criteria that may affect their user experience and purchasing decision were also consulted. This result can help designers to set weights for different aspects of user experience. In doing so, the user evaluation collected can be more reliable.

24.4.3.2 Results and Discussions

As illustrated in the case study, sometimes designers have to conduct user or customer research while the design concepts are not yet mature and physical prototypes are yet to be built. In the case study, the six designs were introduced and presented by PPT slides. Similar to virtual shopping, the sensory experiences are not that complete [53] and users need to interact with the product through limited sensory modalities (mainly relying on visual experience). However, it is believed that the close interrelations and interactions among senses may help to compensate the absence of some stimuli to a certain extent (though it is impossible to fully compensate or replace). For example, the auditory information can be used to improve the visual or tactile perceptions [54–56]. Peck and Childers [57] used "written descriptions and visual depictions of products" to help people obtain tactile information. Thus, designers can still capture valuable potential user experience concerning different aspects of experience if enough information is provided to the users. As shown in Fig. 24.15, one advantage of PPT slides is its convenience and flexibility to better present the designs with images and textual descriptions.

The following discussion is divided according to seven issues, A–G.

Fig. 24.15 Samples of PPT slides in the customer survey

A. **The importance of criteria**
Instead of averaging all criteria, the study takes individual differences into consideration by consulting participants' attitudes towards the importance of different criteria that may affect their experience and purchasing decision. By checking with analysis of variance (ANOVA), it is found that there are no significant individual differences (p-value = 0.817605) between participants. However, there are significant differences on criteria (p-value = 7.19E−31). It is possible that participants shared similar attitudes toward the criteria for this kind of product type—biscuit container. On average, 'functionality' is chosen as the most important criterion, followed by 'appearance', 'emotional experience', and then 'scent'.

B. **Initial user experience**
After viewing and getting to know each design concept, participants were asked to evaluate the product and its potential experience with their first thought during the individual evaluation. Initial user experience is captured by mainly three aspects. First, participants' visual experience with product appearance was examined. As shown in the first part of Table 24.3, the design concept 'Desire' has the highest average ratings compared to the rest, followed by 'Sharkie', which is relatively close to 'Desire'.

Table 24.3 The average ratings of user evaluation on the appearance, functional experience and emotional experience during the individual evaluation

	Mushy	Passion	Desire	Sharkie	Fresh Mint	Mary biscuit	*P*-value
Appearance							
Pleasant	2.65	2.85	**3.13**	*3.05*	2.44	2.65	0.05089
Attractive	2.85	2.84	**3.38**	*3.28*	2.71	2.57	*0.00299*
Modern	2.78	2.98	**3.41**	*3.28*	3.06	2.54	*0.00395*
Interesting	2.80	2.90	**3.40**	*3.21*	2.92	2.63	*0.02838*
Cute	2.88	2.87	**3.48**	*3.24*	2.81	2.55	*0.00186*
Function							
Practical	3.56	3.27	**3.91**	**3.91**	3.40	3.10	*0.00261*
Durable	**3.73**	*3.66*	3.12	*3.68*	3.35	3.35	*0.04447*
Safe	*3.89*	3.47	**3.92**	3.65	3.57	3.75	0.36022
Emotion							
Happy	2.45	2.52	**2.71**	*2.66*	2.26	2.35	0.4364
Inviting	2.46	2.44	**2.83**	*2.65*	2.23	2.16	0.0811
Active	2.22	2.24	**2.63**	*2.52*	2.22	2.01	0.1213
Surprise	2.40	2.51	**2.86**	*2.81*	2.49	2.08	*0.0161*
Satisfied	2.42	2.29	**2.78**	*2.73*	2.34	2.19	0.0730
Comfort	2.40	2.43	**2.70**	*2.53*	2.23	2.31	0.5124

Note The bold and italic values respectively represent the highest and second high average ratings.

Second, the average ratings of functional experience evaluation are shown in the second part of Table 24.3. Different design concepts have no significant effect on the criterion 'safe' but do have effect on the criteria 'practical' and 'durable'. Overall, 'Desire' and 'Sharkie' have the most practical function. 'Mushy', 'Sharkie' and 'Passion' are more durable amongst all the biscuit containers. Lastly, the safety of 'Desire' and 'Mushy' are slightly higher compared to the rest.

Third, emotional experience evaluation is shown in the last part of Table 24.3. The result indicates that different designs of biscuit containers have effects on the 'surprise' criterion and may have effects on the criteria 'inviting' and 'satisfied'. Again, 'Desire' and 'Sharkie' tend to have the highest ratings for all emotional experience evaluations.

C. **Individual differences in olfactory experience**

During individual evaluation, participants chose their preferred type of scents for each design. The result suggests obvious individual differences in personal olfactory habits. For those participants who do not appreciate scents as an attractive design feature chose 'no scent' for most of the designs. For those who enjoy olfactory experience tend to have their own preferred type of scent for different designs. In addition, visual design features tend to have an influence on participants' decision making on the scents. For example, for the design 'Desire', whose shape and colour are directly linked to food—chocolate, 14 participants chose the scent to be chocolate flavor. As for the design 'Fresh-Mint', whose shape and colour resemble tooth and toothpaste, 18 participants chose the scent to be mint. On the contrary, for the design 'Sharkie', whose shape is not really related to food, most participants chose it to be no scent.

D. **Design concepts and the suitable usage occasions**

During individual evaluation, participants chose their preferred usage occasions in which they would like to use the biscuit container. Corresponding to the first stage of the case study, the usage scenarios included both daily use occasions and special events occasions.

The accumulation of chosen times for each design concept and each usage scenario are shown in Table 24.4. The information regarding the design concepts and their suitable usage occasions is valuable for not only designers or product managers, but also for customer segmentation or marketing department. For example, if a company is keen to develop a product or a kind of experience for some specific scenarios or target contexts, this kind of information can help the trade-off in decision making. A company can also explore new usage contexts for a novel product and its experience and then plan for its marketing strategies, e.g., highlight the usage scenario in the advertisement. For instance, if a company wishes to launch a biscuit container during Chinese New Year, the design concept 'Passion' can be a good choice as it shows the highest score for the 'festive seasons'. However, if a company tends to cover more possible usage occasions in order to attract more consumers, the design concepts 'Desire' and 'Sharkie' are more ideal as they are suitable for more usage occasions whether for daily use or special events. From a different point of view, if a company is

Table 24.4 Sum of participants' decision making on the preferred usage occasions

Biscuit container		Mushy	Passion	Desire	Sharkie	Fresh mint	Mary biscuit
Daily use occasions	Breakfast **alone**	10	7	11	11	13	12
	Breakfast **with family**	9	14	12	12	12	12
	Afternoon tea **alone**	14	9	14	14	16	15
	Afternoon tea **with family**	15	17	15	14	12	12
Sum of daily use occasions		48	47	52	51	53	51
Special events occasions	Festive seasons **with companion**	17	26	15	15	7	11
	Exam **alone**	9	9	16	16	17	12
	Movie marathon **with companion**	11	5	13	12	10	10
	Party **with companion**	14	17	16	14	11	9
	Outdoor activities (e.g., picnic) **with companion**	13	15	11	17	8	10
Sum of special events occasions		64	72	71	74	53	52
Sum of all usage occasions		112	119	123	125	106	103

going to launch the product 'Passion', it can highlight its suitability for special events such as festival seasons, e.g., compartments to contain and display different kinds of cookies and sweets, which can elicit a positive feeling of sharing happiness. Designers can also learn from this to know which design features can be in the spotlights and which may be unsuitable for some usage occasions. There is actually much more one can learn and benefit if considering such contextual factors during user experience exploration.

E. **The comparison evaluation**

During the comparison evaluation, participants compared all six design concepts mainly from three aspects. First, participants compared the concepts from a functional point of view. As shown in the first part of Table 24.5, different design concepts have significant effects on the functional experience evaluations. 'Desire' and 'Sharkie' are the most practical designs while 'Mushy' and 'Mary Biscuit' are the most durable and safe designs.

Second, the average ratings of emotional experience evaluation are shown in the second part of Table 24.5. The result suggests that the designs of biscuit containers have effects on all emotional experience evaluations. Similar to the result from individual evaluation, 'Desire' and 'Sharkie' have the highest ratings for all aspects of emotional experience.

Third, participants' willingness to purchase the biscuit containers without considering the cost was consulted during comparison evaluation, which might be a situation closer to the real world, i.e., on the market. As shown in the last part of Table 24.5, the design concepts have significant effects on the participants' purchasing decision. 'Desire' and 'Sharkie' would be purchased by most of the participants, followed by 'Mushy' and 'Passion'.

Table 24.5 The average ratings of user evaluation on functional experience, emotional experience and willingness to purchase during the comparison evaluation

	Mushy	Passion	Desire	Sharkie	Fresh mint	Mary biscuit	*P*-value
Function							
Practical	55.01	53.69	*70.72*	**73.36**	46.15	46.55	8E−06
Durable	*68.58*	46.23	57.05	64.37	44.37	**69.12**	3E−04
Safe	**79.31**	59.92	62.72	45.36	50.26	*72.33*	9E−07
Emotion							
Happy	32.57	35.94	**55.27**	*51.38*	32.05	30.72	4E−07
Inviting	37.90	38.43	**53.44**	*51.37*	29.77	31.65	4E−05
Active	36.09	41.85	*52.29*	**52.70**	39.05	28.23	1E−04
Surprise	29.72	41.85	**56.16**	*55.32*	37.47	26.92	4E−09
Satisfied	35.59	38.94	**55.37**	*52.47*	31.52	32.26	1E−05
Comfort	38.80	40.06	**50.48**	*49.30*	28.07	38.29	1E−03
Willingness to purchase							
	59.03	58.09	**81.24**	*73.15*	49.24	48.67	1.02E−06

Note The bold and italic values respectively represent the highest and second high average ratings.

F. **Qualitative information**

Besides the quantitative data discussed above, the participants' qualitative feedbacks were also collected. On the one hand, qualitative information can help designers have a better understanding of participants' thoughts from varied aspects of user experience rather than be limited by the choices of the answers. It also provides a chance for designers to discover customers' possible doubts and queries, which all can help designers to improve the designs and, thus, more positive experience can be created. For example, participants may query the easiness of cleaning for the brown internal surface of 'Mushy', question whether 'Passion' would fall easily especially when there are kids around, or cannot estimate the cookies' level when it is lower than the position of the eyes of 'Sharkie'. Similarly, participants may commend on some facets of the designs. Some examples are the affective value of the overflow chocolate feature of 'Mushy', which may stir up curiosity and longing; the functional value of the temporary waste storage of 'Desire' especially for outdoor activities; and the sensory design features of the mint scent of 'FreshMint', which may elicit users the feeling of freshness and remind them to brush the teeth. Furthermore, some valuable suggestions can be captured as well. For instance, while commenting on the design 'Sharkie', one participant suggested changing the way of cutting food packaging by sliding the package upwards instead of downwards. This was to avoid the contents inside packaging to leak out during the cutting process.

On the other hand, qualitative information can help designers to assess how well a participant can build a mental model of a design concept in his/her mind. Especially for a virtual situation where users are not able to interact with the physical products, it can be a factor how well a user can imagine a product and his/her potential user experience. Participants may give vivid comments when they are motivated to imagine user experience. For example, one participant noted "*the whole appearance of the stickman makes people happy especially when it looks as if you are trying to steal his food. The tissue paper storage area at the inner section is an added surprise.* [Participant 1—Affective (happy, surprise)]" while evaluating 'Desire'.

G. **The final design concept**

Based on the user experience and evaluation captured, a final design concept is chosen and revised. 'Desire' and 'Sharkie' have the highest voting in most of the aspects. For appearance evaluation, 'Desire' and 'Sharkie' were the most pleasant, attractive, modern, interesting and cute biscuit containers. For functional experience, 'Desire' and 'Sharkie' were selected to be the most practical biscuit containers. For emotional experience, 'Desire' and 'Sharkie' were chosen to be the most happy, inviting, active, surprise, satisfied and comfortable biscuit containers. On top of that, 'Desire' and 'Sharkie' have the highest counts in occasions which participants would use the biscuit containers. 'Desire' and 'Sharkie' once again scored highest in participants' willingness to purchase the biscuit container. Hence, 'Desire' and 'Sharkie' biscuit containers were chosen as the better design concepts among the six. In order to select the final design

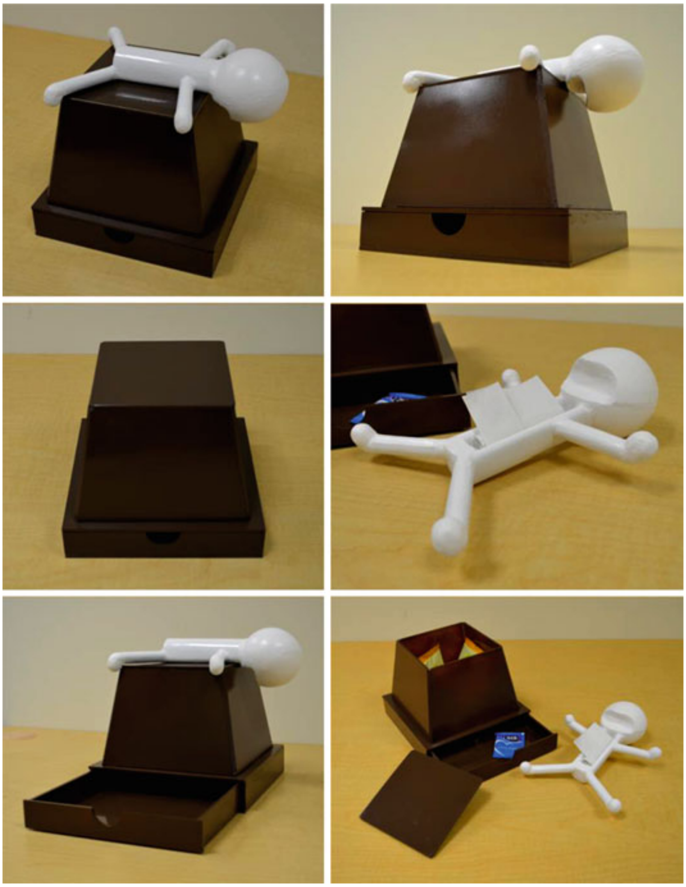


Fig. 24.16 The prototype of the final design, 'Desire'

among them, the qualitative comments given by participants were further examined. It is shown that 'Desire' can evoke more positive emotions of the participants, e.g., happy, surprise, interesting and appetizing emotions. Thus, it is chosen as the final design among the six biscuit containers for this young segment. The design of Desire was revised and improved. Based on the customer requirements captured, the way to open the temporary waste storage is changed to sliding, which is more convenient for users to open the waste storage. This also avoids biscuits crumbled when opening the waste storage. A physical working prototype is built as shown in Fig. 24.16.

24.5 General Discussion

The application of the CMSES and SCS can be tailored readily according to different purposes and needs. A company should first clarify the goal or special interests of its design project. For the case study, the targeted user group is young generation and the product type of interest is biscuit container. The design team is keen to get deeper, more detailed and comprehensive understanding of user experience so as to create more consumer-focused products. As demonstrated at the first stage of the case study, the 'context-based multi-sensory experience exploration' is a promising approach to help designers examine, exploit and investigate user experience in a more dedicated and robust manner.

On the one hand, the study takes individual differences into consideration when tackling multiple contextual factors. This is to ensure the experiential knowledge captured can fit into a user's real situation and, thus, be more valuable and usable. A company can decide how much the users or customers can contribute to the context settings of usage scenario(s), which will be used in the user involvement studies. In the case study, the design team uses an easy and quick way by providing several usage scenarios for participants to choose the ideal one(s) to experience the product. The advantages are fewer budgets and time consuming, as well as no extra training is required from the user side. Nevertheless, for a company that promotes innovation and creativity, users can have more opportunity and freedom to decide the customized scenarios. In doing so, design teams can be better inspired regarding how a product can be 'played' and the marketing department can also explore more potential usage scenarios for advertisement and so on.

On the other hand, due to the fact that users experience a product through all the senses, it is a good starting point for designers to deal with inherently complex user experience in a more natural way. In addition, applying multi-sensory experience exploration is like using a magnifier to examine user experience without missing any sensory aspect in order to have more complete exploration of user experience. For example, design teams can discuss the sensory experience of a certain sense in more detail to identify possible problems caused by a stimulus, and seek the cause and effect of both positive and negative experience so as to explore potential opportunities for 'multi-sensory experience design'. Meanwhile, design teams can probe and examine how the different sensory design features (or stimuli) may affect user experience while working together (i.e., sensory interactions) to ensure the design concept may produce natural, logical and pleasing overall multi-sensory experience. In addition, designers have to ensure whether the design concepts can fit into the targeted usage scenarios after the conceptualization stage. In doing so, the design concepts created can bring users the most long-lasting positive, hedonic and rich multi-sensory experience. Thus, the design concepts generated at the second stage of the case study tend to have better user evaluation than the original design for the young segment as discussed at the third stage of the case study. In this sense, the 'context-based multi-sensory experience design' can also strengthen the experience design.

Moreover, since the knowledge regarding customers' personal sensory preference, sensory habits and ideal usage contexts can be captured, a company can better plan for its marketing and design strategies as well. Yet the application can still be very flexible depending on the project focus. Therefore, one can emphasize more on the contextual factors and individual differences. For example, if a company wishes to develop a product targeting at some specific segments (e.g., the elderly, the 'soho', young parents or athletes), it may start with unearthing what kinds of usage contexts are the most popular for different user groups, followed by deciding the design and marketing strategy. Researchers can also manipulate some contextual factors in the SCS or provide some rules for co-building scenarios with users to investigate some special issues. For instance, a company may give users a premise that the time is 5 years from present and then ask users to build up their imagined future usage contexts. On the other hand, one can focus more on the multi-sensory experience and individual differences. For example, if a company wishes to implement a 'mass customization' strategy in order to expand the customer segments, it can make good use of the knowledge regarding users' individual sensory preference. As shown in the case study, participants tend to have stronger individual differences on the olfactory and auditory experiences. Accordingly, designers can provide more diverse sensory design features (e.g., more choices on the scents for such biscuit container design) to satisfy different customer segments while simultaneously control the manufacturing costs. As a result, in a CE/CPD environment, different departments in a company can work together better.

In a sense, the study may consume more time and cause higher budget because it suggests more input from the user or customer domain. However, it is still worth endeavor and investment for a company to implement, especially when the consumers' experience has become more and more important in this highly competitive era. In addition, this concept can be carried out at the very front end of PDD to maximise the benefits.

24.6 Conclusion and Future Perspectives

As the contemporary PDD process has shifted the focus of its endeavour to user-centric and consumer-oriented aspects, users or customers are often invited to contribute their views in NPD. Nevertheless, some issues would rise readily if there is lack of careful consideration of the inherent nature of user experience during user involvement studies. Based on the identified research gaps, a prototype CMSES with a SCS is introduced and demonstrated using a three-stage case study. The result is promising and shows valuable potential benefits for a company to employ the proposed 'context-based multi-sensory experience exploration and design' approach in designing and developing consumer goods. It is envisaged that, with the proposed approach, not only designers can have a more comprehensive and in-depth understanding of users' multi-sensory experience, but also product managers and marketers can better plan for the design and marketing strategies. This is because valuable knowledge regarding individual differences in different aspects of user experience, including personal sensory habits and ideal usage contexts, can be captured.

User experience has become a crucial key to success in designing and developing consumer goods. Future studies can dedicate to investigate into different methodologies and tools to address the complex, subjective and dynamic nature of user experience and strengthen user experience exploration [58, 59]. More empirical studies are also required to demonstrate the proposed 'context-based multi-sensory experience exploration and design' approach and broaden the application scope by inviting more stakeholders to facilitate successful implementation of CE/CPD in the realm of PDD [60].

References

1. Haque BU, Belecheanu RA, Barson RJ, Pawar KS (2000) Towards the application of case based reasoning to decision-making in concurrent product development (concurrent engineering). Knowl-Based Syst 13(2–3):101–112
2. Riedel JCKH, Pawar KS (1991) The strategic choice of simultaneous versus sequential engineering for the introduction of new products. Int J Technol Manage 6(3–4):321–334
3. Swink ML (1998) A tutorial on implementing concurrent engineering in new product development programs. J Oper Manage 16(1):103–116
4. Addo-Tenkorang R (2011) Concurrent engineering (CE): a review literature report. Paper presented at the world congress on engineering and computer science (WCECS 2011), San Francisco, USA, 19–21 Oct
5. Winner RI, Pennell JP, Bertrand HE, Slusarczuk MMG (1988) The role of concurrent engineering in weapon system acquisition (vol IDA report R-338). Institute for Defense Analyses, Alexandra
6. Shishko R (ed) (1995) NASA systems engineering handbook (vol. SP-610S). National Aeronautics and Space Administration, Washington
7. Chen CH, Sato K, Lee KP (2009) Human-centered product design and development. Adv Eng Inf 23(2):140–141
8. Sanders EBN (1999) Postdesign and participatory culture. Paper presented at the useful and critical: the position of research in design, Tuusula, 9–11 Sept
9. Nagamachi M (2002) Kansei engineering as a powerful consumer-oriented technology for product development. Appl Ergon 33:289–294
10. Chen CH, Yan W (2008) An in-process customer utility prediction system for product conceptualisation. Expert Syst Appl 34(4):2555–2567
11. Hende EA, Schoormans JPL (2012) The story is as good as the real thing: early customer input on product applications of radically new technologies. J Prod Innov Manage 29(4):655–666
12. Yan W, Chen CH, Chang W (2011) A functional—commercial analysis strategy for product conceptualization. Expert Syst Appl 38(8):9879–9887
13. Carbonell P, Rodriguez-Escudero AI, Pujari D (2009) Customer involvement in new service development: an examination of antecedents and outcomes. J Prod Innov Manage 26(5):536–550
14. Pine BJ, Gilmore JH (1998) Welcome to the experience economy. Harvard Bus Rev 76(4):97–105
15. Pine BJ, Gilmore JH (1999) The experience economy: work is theatre and every business a stage: goods and services are no longer enough. Harvard Business School Press, Cambridge

16. Boven LV, Gilovich T (2003) To do or to have? That is the question. J Pers Soc Psychol 85 (6):1193–1202
17. Djajadiningrat JP, Overbeeke CJ, Wensveen SAG (2000) Augmenting fun and beauty: a pamphlet. Paper presented at the DARE 2000: designing augmented reality environments, Helsingor, 12–14 Apr
18. Gilovich TJCT (2010) The Relative Relativity of Material and Experiential Purchases. J Pers Soc Psychol 98(1):149–159
19. Hassenzahl M, Tractinsky N (2006) User experience—a research agenda. Behav Inf Technol 25(2):91–97
20. Desmet P, Hekkert P (2007) Framework of product experience. Int J Des 1(1):57–66
21. Hekkert P, Schifferstein HNJ (2008) Introducing product experience. In: Schifferstein HNJ, Hekkert P (eds) Product experience. Elsevier, Amsterdam, pp 1–8
22. Stappers PJ, Hv Rijn, Kistemaker SC, Hennink AE, Visser FS (2009) Designing for other people's strengths and motivations: three cases using context, visions, and experiential prototypes. Adv Eng Inform 23(2):174–183
23. Buchenau M, Suri JF (2000) Experience prototyping. Paper presented at the 3rd conference on designing interactive systems: processes, practices, methods, and techniques, Brooklyn, New York
24. Rodrigues C, Hultén B, Brito C (2011) Sensorial brand strategies for value co-creation. Innovative Mark 7(2):40–47
25. Law ELC, Roto V, Hassenzahl M, Vermeeren APOS, Kort J (2009) Understanding, scoping and defining user experience: a survey approach. Paper presented at the 27th international conference on human factors in computing systems, Boston, 4–9 Apr
26. Chen NF, Chen CH, Khoo LP Foo C (2012) investigation into dynamic multi-sensory product experience based on online shopping. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, (2013). Proceedings of the 19th ISPE international conference on con-current engineering. Springer, London, pp 897–908
27. Chen NF, Ho CH, Ma MY (2011) Sensory importance and emotions at early stage of product experiences—a qualitative study of juice squeezer. Paper presented at the 5th conference on designing pleasurable products and interfaces, Milan
28. Fenko A, Schifferstein HNJ, Hekkert P (2010) Shifts in sensory dominance between various stages of user-product interactions. Appl Ergon 41:34–40
29. Jakesch M, Zachhuber M, Leder H, Spingler M, Carbon CC (2011) Scenario-based touching: on the influence of top-down processes on tactile and visual appreciation. Res Eng Des 22 (3):143–152
30. Hummels C, Djajadiningrat J, Overbeeke C (2001) Knowing doing and feeling : communicating with your digital products. Paper presented at the Interdisziplinäres Kolleg Kognitions und Neurowissenschaften, Günne am Möhnesee, 2–9 Mar
31. Hekkert P (2006) Design aesthetics: principles of pleasure in design. Psychol Sci 48(2):157–172
32. Huang Y, Chen CH, Khoo LP (2012) Products classification in emotional design using a basic-emotion based semantic differential method. Int J Ind Ergon 42(6):569–580
33. Norman, DA (2004) Emotional design: why we love (or hate) everyday things. Garden City Publishing Ltd
34. Chen CH, Khoo LP, Yan W (2006) An investigation into affective design using sorting technique and Kohonen self-organising map. Adv Eng Softw 37(5):334–349
35. Khalid HM, Helander MG (2006) Customer emotional needs in product design. Concurrent Eng Res Appl (CERA) 14(3):197–206
36. Huang Y, Chen CH, Khoo LP (2012) Kansei clustering for emotional design using a combined design structure matrix. Int J Ind Ergon 42(5):416–427
37. Lee S, Harada A, Stappers PJ (2002) Design based on Kansei. In: Green WS, Jordan PW (eds) Pleasures with products: beyond usability. Taylor and Francis, London, pp 219–230

38. Nagamachi M (1995) Kansei engineering: a new ergonomic consumer-oriented technology for product development. Int J Ind Ergon 15:3–11
39. Schifferstein HNJ, Cleiren MPHD (2005) Capturing product experiences: a split-modality approach. Acta Psychol 118:293–318
40. Schifferstein HNJ, Spence C (2008) Multisensory product experience. In: Schifferstein HNJ, Hekkert P (eds) product experience. Elsevier, Amsterdam, pp 133–161
41. Spence C (2011) Managing sensory expectations concerning products and brands: capitalizing on the potential of sound and shape symbolism. J Consum Psychol 22(1):37–54
42. Bloch PH (2011) Product design and marketing: reflections after fifteen years. J Prod Innov Manage 28(3):378–380
43. Lindstrom M (2005) Brand sense: build powerful brands through touch, taste, smell, sight, and sound. Free Press, New York
44. Schifferstein HNJ, Desmet PMA (2007) The effects of sensory impairments on product experience and personal well-being. Ergonomics 50:2026–2048
45. Holmes NP, Sanabria D, Calvert GA, Spence C (2007) Tool-use: capturing multisensory spatial attention or extending multisensory peripersonal space? Cortex 43(3):469–489
46. Zomerdijk LG, Voss CA (2011) NSD processes and practices in experiential services. J Prod Innov Manage 28(1):63–80
47. Edvardsson B, Olsson J (1996) Key concepts for new service development. Serv Ind J 16 (2):140–164
48. Rijn Hv, Stappers PJ (2008) Expressions of ownership: motivating users in a co-design process. Paper presented at the 10th anniversary conference on participatory design, Bloomington
49. Go K, Carroll JM (2004) The blind men and the elephant: views of scenario-based system design. Interactions 11:44–53
50. Schifferstein HNJ (2006) The perceived importance of sensory modalities in product usage: a study of self-reports. Acta Psychol 121:41–64
51. Spence C (2002) The ICI report on the secret of the senses london: imperial chemical industries plc. The Communication Group, London
52. Schifferstein HNJ, Desmet PMA (2008) Tools facilitating multi-sensory product design. Des J 11(2):137–158
53. Citrin AV, Stem DE, Spangenberg ER, Clark MJ (2003) Consumer need for tactile input: an internet retail challenge. J Bus Res 56:915–922
54. Durlach NI, Mavor AS (eds) (1995) Virtual reality: scientific and technological challenges. committee on virtual reality research and development. National Research Council, Washington
55. Klatzky RL, Pai DK, Krotkov EP (2000) Perception of material from contact sounds. Presence 9(4):399–410
56. Kunkler-Peck AJ, Turvey MT (2000) Hearing shape. J Exp Psychol Hum Percept Perform 26 (1):279–294
57. Peck J, Childers TL (2003) To have and to hold: the influence of haptic information on product judgments. J Mark 67(2):35–48
58. Chang D, Chen CH (2014) Understanding the customer involvement in radical innovation. In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 72–80
59. Chang D, Chen CH (2014) Understanding the influence of customers on product innovation. Int J Agile Syst Manag 7(3/4):348–364
60. Chang D, Chen CH (2014) Exploration of a concept screening method in a crowdsourcing environment. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 861–870

Chapter 25
The Application of an Integrated Product Development Process to the Design of Medical Equipment

Osiris Canciglieri Junior, Maria Lucia Miyake Okumura and Robert Ian Marr Young

Abstract With the research presented in this chapter we aim to investigate the importance of the concurrent engineering (CE) philosophy in the engineering-medical multidisciplinary environment for integrated product development process (IPDP) of medical equipment. We address the requirements of a health professional user as well as patient's needs. We have identified and contextualized the medical equipment lifecycle, the importance of CE in the IPDP of medical equipment and present propositions for the insertion of software tools that support product development phases. A discussion is included on the use of CE and IPDP oriented towards medical equipment conception and development, perspectives of engineering modular development and interface between Health and Engineering information areas for increasing technical, clinical and economic quality.

Keywords New product development · Virtual product creation · Concurrent engineering · Systems engineering · Product lifecycle management

O. Canciglieri Junior (✉) · M.L.M. Okumura
Polytechnic School - Production and System Engineering Graduate Program - PPGEPS, Pontifical Catholic University of Paraná (PUCPR), R. Imaculada Conceição, 1155, Paraná 80215-901, Brazil
e-mail: osiris.canciglieri@pucpr.br

M.L.M. Okumura
e-mail: lucia.miyake@pucpr.br

R.I.M. Young
Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Ashby Road, Loughborough, Leicestershire LE11 3TU, UK
e-mail: R.I.Young@lboro.ac.uk

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_25

25.1 Introduction

Concurrent engineering (CE) has been increasingly used within the product development process (PDP) to support the searching for innovation, strategic alternatives and smart solutions to reduce the cost and lead-time. In this sense, the integrated product development process (IPDP) within a CE environment associated to the Health Technology could face the challenge of developing products that meet specific and customized conditions integrating multidisciplinary areas in the design phases.

Currently, most medical tools try to promote patient's quality of life through faster and more accurate diagnostics and less invasive screening and diagnostic tests, depend on high technology equipment. These medical equipment expose the close connections between engineering and medical areas since the new ideas planning must integrate the whole structure of the PDP. In this context, the research's objective present in this chapter, investigates the importance of the CE philosophy in the engineering-medical multidisciplinary environment for IPDP of medical equipment, addressing the requirements of the health professional user as well as the patient's needs. The methodology applied is theoretical with a qualitative approach and experimental nature. The study identifies and contextualizes the medical equipment life cycle, the importance of CE in the IPDP of medical equipment and presents propositions for the insertion of software tools that support the product development phases.

In the Sect. 25.2 medical equipment design oriented to IPDP is introduced. Its relationship to CE environment is described in Sect. 25.3. Discussion on the use of CE and IPDP is given in Sect. 25.4. Three cases studies are presented in Sect. 25.5 to illustrate the whole development process and the value of the proposed methodology. At the end, Sect. 25.6 presents final considerations on the use of CE and IPDP oriented for medical equipment conception and development, the perspectives of engineering modular development and the interface between Health and Engineering information areas seeking an increase in the technical, clinical and economic quality.

25.2 Medical Equipment Design Oriented to Integrated Product Development Process

This topic first contextualizes the medical devices as well as their definition, function and classification, and then addresses the process of developing new products targeted for medical area and its evolution/innovation. At the end, it is presented the importance of CE philosophy in the integrated development process of medical devices.

25.2.1 Medical Equipment and Classification

The Brazilian Health Surveillance Agency—ANVISA [1] defines medical equipment through the RDC 2/2010 Norm as “equipment or system, including its accessories and parts for medical use or application, dentistry or laboratory, used direct or indirectly for diagnostic, therapy and monitoring in the population health care, without using pharmacological, immunological or metabolic means to perform its main function in human beings, which may, however, be assisted by such means”. WHO [2] considers a medical equipment as any instrument, apparatus, machine, implant, and similar applied to humans, according to the functions described as follows: (a) diagnosis, prevention, monitoring, treatment or alleviation of disease; (b) diagnosis, monitoring, treatment, alleviation of or compensation for an injury; (c) investigation, replacement, modification, or support of the anatomy or of a physiological process; (d) supporting or sustaining life; (e) control of conception; (f) disinfection of medical devices; and (g) providing information for medical purposes by means of in vitro examination of specimens derived from the human body and which does not achieve its primary intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means.

The Food and Drug Administration has about 1,700 different types of generic medical devices classified and established, which are grouped in 16 specialties [3]. These devices are categorized into: pre-amendment devices, post-amendment devices, substantially equivalent devices, implanted devices, custom devices, investigation devices and transitional devices. Nevertheless, WHO [4] estimates around 10,000 types of medical devices available worldwide, not counting its variants. In 2008, this sector generated approximately US$210 billion employing over 1 million people in 27,000 companies located around the world. The projection for this sector is an economic growth of around 6 % per annum.

Classification of medical devices is related to the operational complexity degree of the equipment, the cost and time needed for training the health professional. According to Calil and Teixeira [5] this complexity can be: (a) low-easy operation, no need for skilled human resource and training; (b) medium-requires human resources with basic formation and equipment proper training; and (c) high-demands qualified and specialized technicians and specific training.

25.2.2 Development Process of Medical Equipment

The development of new products in the medical field is the result of technological advances enabling an improvement in people’s lifetime since diagnoses and treatments of diseases are earlier and more efficiently detected. However, the reality of new ideas and technologies associated with health, in the proposed and developed products, presents a success probability of only one case out of a hundred [6].

The level of complexity in the development stages of medical equipment is increasing due to the demand for more accurate and effective examinations and therapies. The challenge lies especially in the technological mastery of several knowledge areas for the holistic design of the required concept.

25.2.3 Medical Equipment: Evaluation and Innovation

Currently, medical equipment is present in most clinical activities, such as the preventive tests, during treatment or monitoring, rehabilitation therapies and it is fundamental in the medical field, but its acquisition requires considerable investment because of its initial price, high cost of maintenance, and also by rapid technological obsolescence [7, 8]. This scenario led managers to adopt more systematic and rational health assessment processes for the acquisition of new equipment, while also considering safety, efficiency, quality and good manufacturing practices. Amongst the assessment processes, there is the health technology assessment (HTA) that started with the computer aided tomography advances aiming to absorb low cost new technologies and seeking for better cost-efficiency [9, 10]. Figure 25.1 illustrates the HTA behavior during the product life cycle and shows that HTA rate increases according to the medical equipment usage in function of time. It is noticed that there are a small number of users of medical

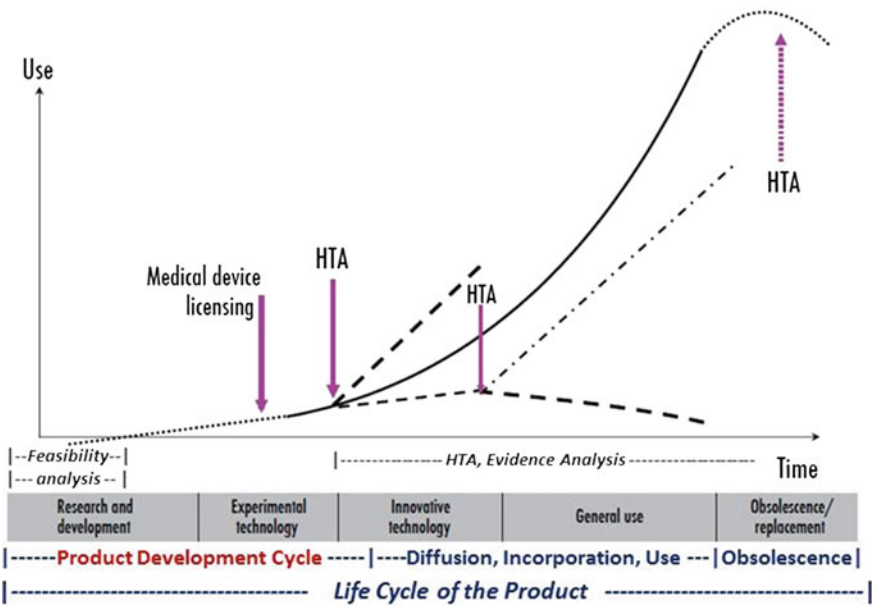


Fig. 25.1 Health technology assessment, diffusion of health technologies, feasibility analysis and evidence analysis in the research and product development. Based on [6, 8, 10, 19]

equipment with low rate of HTA during the product development cycle. However, this number of users can increase or decrease during the diffusion and incorporation of the product due to different obtained performance, shown in the figure by HTA rate forking which highlights the adherence of future users. Thus, as the analyses of the evidence is occurring and bringing good results, the rate of hypertension is likely to increase. This growth can be evidenced even in the stage of obsolescence of medical equipment. Therefore, it appears that the rate of assessment is low in launching products, it is advisable to do a deeper analysis of technical and economic feasibility since the beginning of the planning phase of the IPDP [6, 11].

Another assessment process is the evidence-based medicine (EBM), which uses tools of clinical epidemiology, statistics, scientific methodology and informatics to build knowledge and achieve higher performance applied to healthcare. The EBM objective is to provide information to support decision making in clinical practice. The Cochrane Center of Brazil provides a database of systematic review of SBE allowing information exchange free of charge worldwide. This database contains information on the COCHRANE analyzes occurrences from documented evidence provided during the practice of the process of medical care worldwide [12, 13]. Thus, medical device evaluation is related to technological innovation according to the formalities or requirements of the medical area, integrating new technologies, processes and organizational environments that result in the advancements on the medical field and contribute to the health of society [10, 14–18].

25.2.4 Concurrent Engineering in the Integrated Development

According to [20–23] the main features of CE are: (a) the emphasis on customer satisfaction; (b) the activities of multidisciplinary teams; (c) the autonomy of teams; (d) the simultaneous development; (e) the leadership to coordinate the entire process of product development; (f) the designs standardization; (g) the information sharing; (h) the use of computerized tools to streamline the processes; and (i) the management practices and instruments for quality assurance. The environment of CE ensures the product quality during the IPDP phases in the life cycle of health care products, leading to a reduced launch time and development costs since the CE environment allows the creation of the concept using various areas of knowledge, understanding more clearly the planning aspects throughout the IPDP phases [11, 24–26]. Decisions made at the beginning of the development cycle are responsible for approximately 80 % of the product final cost [27].

The development should be oriented to the design planning during the IPDP phases, using engineering systematic methods with interdisciplinary approaches, as well as providing flexibility to the integrated environment seeking the best alternatives to meet the needs of consumers. CE is a philosophy that must be applied to the IPDP taking into consideration the requirements of the users and also the demands of the market through the use the tools, models, methods and concepts to identify the best alternative [28]. Therefore, the agents involved when using CE are known as the Seven Ts-Tasks, Teamwork, Techniques, Technology, Time, Tools and Talents [29]. Among these, the following resources stand out: quality function deployment (QFD), Computer-aided-design/Computer-aided-manufacturing (CAD/CAM), Design for Configuration, Design for Precision, Design for Aesthetics, design for safety/liability (DFS), design for life cycle (DFLC), design for manufacturing (DFM), design for assembly (DFA), design for reliability (DFR), Design for Use/Ergonomics/Human Factors, Universal Design, and so on [52, 53].

25.3 IPDP of Medical Equipment in a CE Environment

In the IPDP of medical equipment the users' values and demands and adverse external factors are included, which may relate directly or indirectly to the product development cycle, such as the discovery of new diseases or epidemiological changes, political and/or economic demands, or even demographical aspects. Nonetheless, these demands contribute to the medical field driving technological innovation and bringing new challenges in the search for solutions that can be deployed in the improvement of the functions of the medical equipment.

However, the improvement of the innovation performance lacks a conceptual framework that is able to describe the process as a whole and link knowledge with the skills and competencies necessary to control and strategically manage the product development cycle. A holistic framework would present reliability and a common language for advanced discussions and for information exchange by the teams involved in the process [32]. In this way, CE in IPDP promotes a robust involvement of multidisciplinary areas, which in this research are the engineering and the medicine, incorporating clinical, technical, operational and economic factors that ensure innovation or product evolution. It is worth mentioning that different areas influence the process, according to the competence granted in each IPDP phase, as illustrated in Fig. 25.2. The figure illustrates the PDP that consists of three stages: (a) pre-development; (b) development; and (c) post-development. In the initial stage process the medical field predominates, then decreasing during the IPDP phases and gradually attributing competence for the engineering area. Thus, after the stage of IPDP, the product should pass through the implementation, production and market launch process respectively. This period of post release is relevant because the information feedback from the market and especially the consumer's voice provides information that will serve as opportunities for improvement and product innovation [33]. This procedure also enables the improvement of product quality during the development cycle of the product.

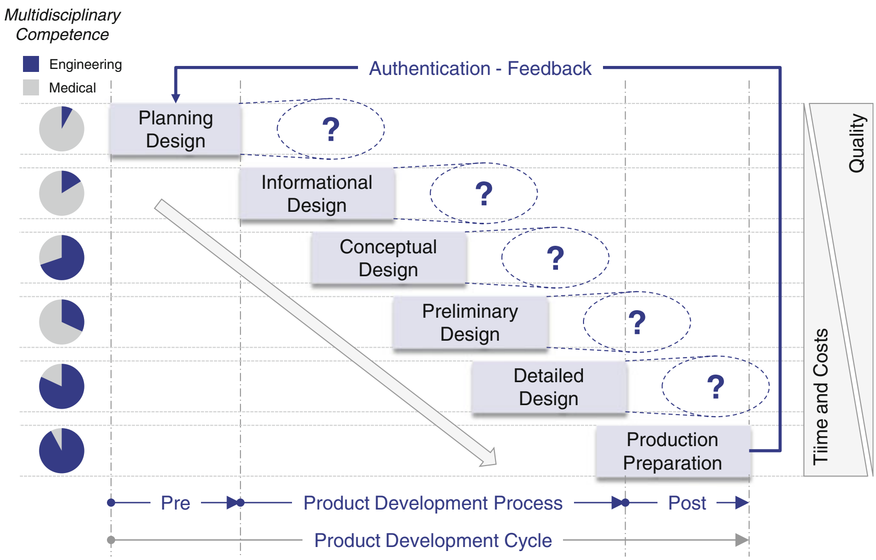


Fig. 25.2 Concurrent engineering in the integrated product development process of medical equipment

25.3.1 Medical Equipment Users

Medical equipment aims to meet the different expectations of their users. First, there are the health professionals who use or handle the equipment and need the knowledge about the equipment's operation and maintenance. Second, there are the patients who are the direct user of the equipment and are under the care of a health professional for diagnosis or treatment.

For the professional who handles the equipment, the requirements are primarily: practicality, comfort and flexibility of use, facilitating learning and ensuring the service quality to the patient, offering confidence in diagnostic procedures or treatment. From the patient point of view, the requirements focus on the comfort aspects while using the equipment and trusting in the procedures results. Health professionals seek to improve the patients' quality of life through the adequate examination procedure or treatment selection. In this context, the principles of Universal Design and Assistive Technology products provide the flexibility to meet the majority number of users and contribute to the mobility aspect of the patient with some limitation or disability [34]. Analogously, the conditions of equipment usage for the professional user such as ergonomics and usability must be considered [16, 17].

25.3.2 Pre-development Phase and Design Feasibility

The pre-development stage is the design planning, which consists of gathering relevant information from the medical area. According to [11, 24] this information is related to the design and product scope such as: (a) in the prognoses of the activities and their duration, deadlines, budgets; (b) in the definition of the responsible team; (c) in the resources needed to undertake the design; (d) in specifying the criteria and procedures for evaluating the quality; and (e) in the risk analysis and performance indicators selected for the design and product.

The Design Planning phase concentrates on the clinical information and techniques to investigate the user's need and market's demand and also to gather the first required characteristics of the product. CE investigates the user's requirements with a multidisciplinary view, identifying needs and creating opportunities for innovation and product evolution. Thus, issues of demands and market competitiveness are incorporated in the feasibility analysis that shows the design stability in the development cycle of the product.

25.3.3 IPDP Phases and Demand Factors

The design elaboration is a macro phase of Product Development and consists of the Informational Design, Conceptual Design, Preliminary and Detailed Design phases. These phases focus on the task set that is characterized by design scope, execution time, resources and risk prediction. The results of these phases respectively yield: design specifications, product conception, the technical and economic feasibility, and documentation. Some considerations for each design phase are:

1. **Informational Design Phase**: product design specifications, in which the needs of users are identified while considering different attributes: functional, ergonomic, safety, reliability, modularity, aesthetic and legal, among others [11]. This information comes from users of medical equipment. The activities of this phase include the survey and description of the tasks and actions that reveal the conditions and activity execution manner by the user [34–36].
2. **Conceptual Design Phase**: the objective is to establish the product functional structure contemplating the definition of the global function. During this phase the marketing planning has the task to monitor the market and identify the changes that may influence the development. Thus, the desirable characteristics of products is investigated considering the interactions with the users (health professional and patient), taking into account the balance of the technical qualities, ergonomic and aesthetic characteristics [37]. These features may be detailed by applying tools such as failure mode and effect analysis (FMEA) that identifies the most important component of the equipment or by using Value Analysis to determine the essential parts of the product.

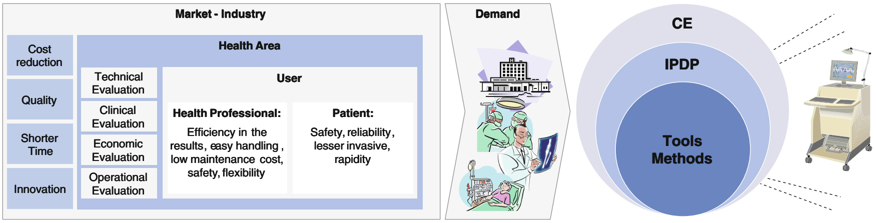


Fig. 25.3 Holistic view of medical equipment development

3. **Preliminary Design Phase**: the development team starts the design planning with updating and understanding the design specifications concerning the form, material, safety, ergonomics, manufacturing and others. At this stage, the prototype is elaborated by using resources such CAD/CAM environments that allow the three dimensions of design configuration to present practical, rapid and inexpensive solutions in the IPDP increasing product quality [38–40].
4. **Detailed Design Phase**: in this stage, the conceptual design is detailed and documented according to the engineering standards and the technical specifications of medical equipment. This detailed design is the basis for the manufacturing planning and the production line costs survey [41].

Each design phase requires strategic planning in order to find the best solutions considering the specificity of users and medical equipment. These specificities are obtained by using the design process support tools such as QFD, FMEA, Ergonomics and Usability, as illustrated conceptually in Fig. 25.3. In this context, the requirements for assembly process simplification, the reduction on the number of components, the parts standardization and the handling facilitation can be implemented using tools such as DFA, DFM, DFS, which leads to a reduction of time and costs for product manufacturing.

Therefore, during the phases of product development the more information and experiences are introduced, formalized and systematized in the processes, the higher will be the level of maturity of the IPDP, which will be reflected in the quality of the final product [42].

25.3.4 Post-development Phase and the Introduction of the Equipment on the Market

In Post-development stage, preparations for production and implementation of marketing planning is made, including the elaboration of assembly documentation, clearance for tooling construction, factory floor preparation, implementation of the production line, among others [11, 34, 41]. Thus, this phase comprehends the evaluation of product performance, comparison and adjustments in the technical specifications. Also, the after-sales follow up services is monitored and the learning and experiences documentation for the subsequent designs is produced, what serves as a feed-back for the IPDP with the positive results, discarding the negative ones [42].

Thus, this phase is the transition from the area of engineering to the medicine, which will introduce and incorporate the medical equipment on the market. It would make the products more safety, use and maintenance technical specifications relevant and produced regarding the language for the health professional user and associated with the clinical and technical validation that results in the product performance [15, 18, 30].

25.4 Discussion on the Use of CE and IPDP Oriented to Medical Equipment

The technological level of medical equipment is in a continuous advancement due to the integration of engineering and medicine areas that contributes to success of the IPDP. The IPDP is supported by the CE environment, which manages the product development phases simultaneously, according to the expertise of each product segment. These segments are explored in IPDP through the application of engineering tools with the goal of finding the best solution that can be implemented in a modular manner (Product Family) allowing the rationalization and optimization of the IPDP phases.

However, there is a need to develop models of tools specifically oriented for the medical area, which interacts properly between areas to absorb the demands of the market. “Getting the design right means exhaustively investigating the product’s functional requirements and the users’ needs and preferences” [43]. So, further detailed studies are necessary and possible adjustments in the methods and existing tools need to be investigated to achieve a higher level of reliability and efficiency of the medical equipment [30, 44].

This association between medicine and engineering also contributes to aspects of integration interfaces in medical equipment that needs to be compatible with each other. Therefore, in view of an overall plan of the CE-oriented medical equipment, as illustrated in Fig. 25.4, the gray arrows represent the development of equipment linked directly with the information and concepts in the medicine field, which are integrated with engineering tools in the IPDP. However, there is the possibility of using this part of this knowledge to develop other products related to this area, which are showed by the white arrows [45]. Moreover, there are opportunities for new product development in different fields [46].

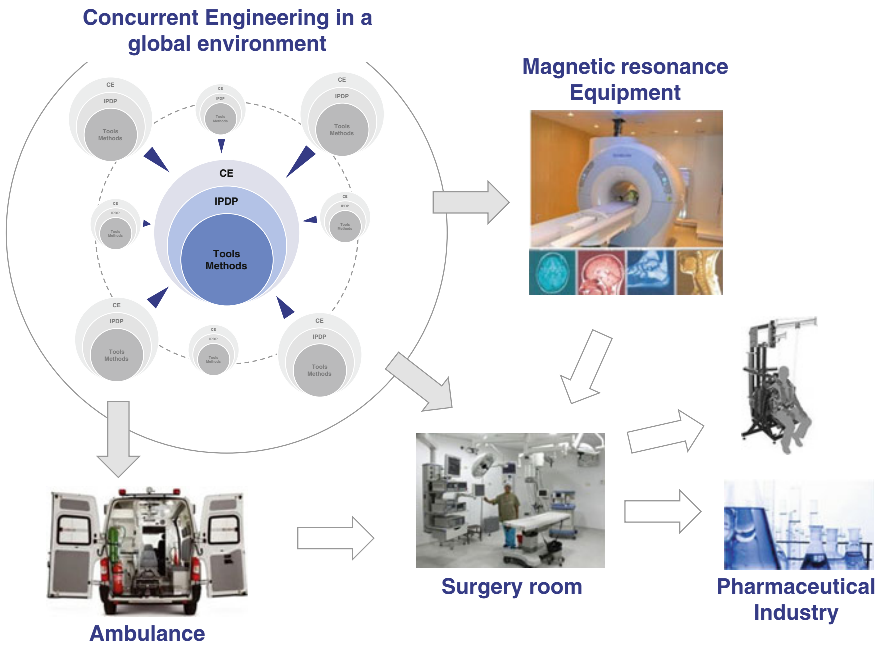


Fig. 25.4 The importance of CE in the global environment of medical equipment

25.4.1 CE and the Interfaces Between Medical Equipment

CE influences the design process through the correct dimensioning of the medical equipment conceptual structures that will be in activity jointly or in sequence of use. The IPDP structure must predict the equipment's joint activities or sequences for a good performance. In this type of interface, CE takes direct action on design activities since it allows a wide view on the operation and handling of the functions of a multi-functional equipment or even different equipment working simultaneously or sequentially in the same environment [25].

A quick and efficient operation between the equipment and/or within the same equipment depends on the professional user's decision and on the handling skill of these health processes. Thus, the user interface makes relevant the tasks and actions performed survey and the searching for interaction solutions that facilitate handling or rationalize the movements necessary to optimize the overall lead-time of their activity, safely and accurately.

The possibility of connection between medical equipment allows the appliance of auxiliary tools to complement the function or even to flexibly use different equipment, which can be selected by the professional user according to the necessity. This flexible characteristic aims to achieve adaptations and connections, which are designed in CE to develop the configuration of accessories of the equipment in the IPDP. This configuration requires a standardization of connection parts targeting at the accessory use in different equipment or functions. Thus, in the CE environment the DFM/DFA/DFX tools can be applied with the universal principles aspect. That means considering the perspectives for the equipment easy assembly and disassembly, use and maintenance [30, 31]. The evolution of the product allows that the accessory with the highest frequency of use to be incorporated into the main unit, or even expand its functions to perform other activities in accordance with market trends.

The CE environment, because of its multi-disciplinary character, enables the interaction among distinct but correlated areas such as diagnostic medicine, rehabilitation medicine, adapted vehicles and pharmaceutical industry, among others. The proposed multidisciplinary environment for medical equipment becomes an important tool to extend the solutions, technically strengthening the IPDP and bringing improvement in technical, clinical and economic product quality.

The information, which is compatible and shared among areas, enables a deeper understanding of each field. Yet, few medical devices with the structure to encode and share information are available. Thus, there is a necessity to develop interfaces that allow compatibility among tools and methods from different areas so that, the equipment can fully concentrate the areas involved in IPDP adding clinical and technical value besides exchanging experiences, knowledge and practices [47].

25.5 The Application of IPDP Concepts on the Development of Medical Equipment—3 Preliminaries Cases Studies

This topic presents three cases studies in order to illustrate the Development of Medical Equipment throughout of IPDP concepts. The first case study explores the design methodology for geometric modeling of customized prosthesis; the second is to show the design and development of a conceptual system and device for acquisition, conversion, transmission and processing of biomedical data; and the third to illustrate a methodology to determine a suitable implant for a single dental failure.

25.5.1 Design Methodology for Geometric Modeling of Customized Prosthesis

There is a substantial need for reconstructive or replacement surgery worldwide due to mainly car accidents, accidents at the work, radical sports, genetically-based malformation and pathological and degenerative illness. The current bone replacement methods use the selection of an approximate prosthesis from a preformed selection normally made out of metal, high density polyethylene or

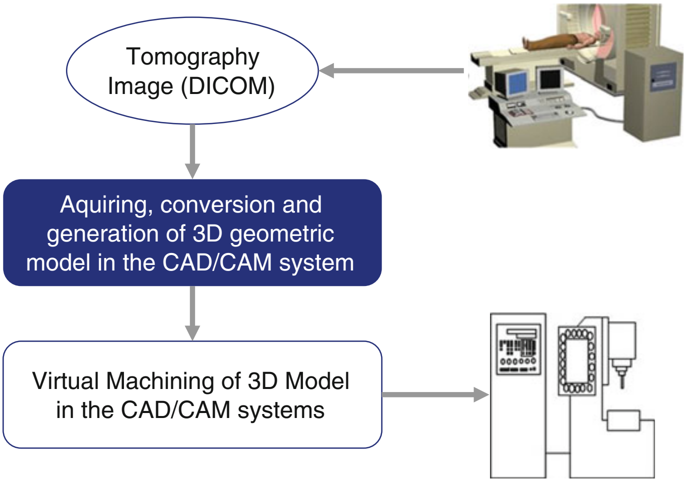


Fig. 25.5 Conception and manufacturing of prosthesis models

ceramics using the surgeon expertise or hand-making an appropriate prosthesis. It requires adjustments during the surgery process that increase substantially the surgery time and costs as well as can cause, in certain cases, traumas to the patient.

Canciglieri et al. have developed a methodology for modeling geometrically custom design prosthesis for surgical use through mathematical extrapolation of data from digital images obtained via tomography of individual patient's bones [48, 49]. Individually tailored prosthesis designed to fit particular patient requirements as accurately as possible should result in more successful reconstruction, enable better planning before surgery and consequently fewer complications during and after surgery. The methodology consists of two main functions as shown in Fig. 25.5. The first function, the focus of this research, is the acquirement and conversion of the tomography image to produce a 3D geometric model in CAD system (see Chap. 12), and the second one is the virtual machining of the 3D model generated in the first function.

To demonstrate the feasibility of virtual model creation, it was used part of a cow femur bone once it was easily purchased from a specialist supplier. The selected part allowed about 288 tomography cuts to be taken by a helicoidally tomography scan machine from General Electric Medical Systems, model HiSpeed CT. The cut analysis interval started at the CUT 100 and finished at the CUT 140. The procedures to obtain the cloud of points were: (a) removal of unwanted lines and imperfections due to the bone porosity; (b) removal of the internal edge; (c) edge detection techniques; (d) image preparation for Cartesian coordinate's acquisition; and (e) the acquisition of the Cartesian coordinates.

The data, after converted from pixels to mm, were imported into the CAD system and the repositioned planes were used in the loft command. The 40 cuts used for the reconstruction are shown in Figs. 25.6 and 25.7 is depicted the 3D solid model obtained after the appliance of the loft command in CAD system.

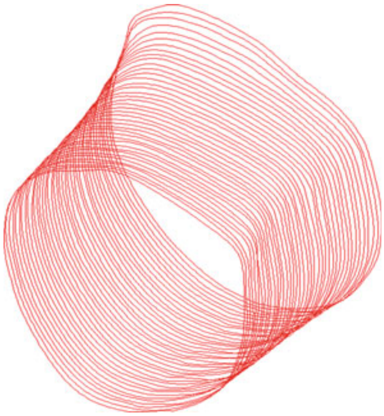


Fig. 25.6 Cuts positioned for the application of loft technique

Fig. 25.7 Application of loft command (to the parallel profiles)

To prove the dimensional results of the virtual model, the tomography femur was dimensioned. After the interval of interest was determined, three marks were made to prove the dimensioning (cuts 100, 120 and 140). The dimensioning was made through contact, at each mark, of the pointer of a 3-dimensional coordinates measure machine (DEA—Sirroco) with a certainty of 3 microns. The virtual model obtained by the methodology and the virtual models obtained by the 3-dimensional measure machine were compared at eight referential points to verify their accuracies. The results showed that, although there are some divergences points, for all three marks, the curves are very similar and overlap which effectively demonstrated the feasibility of the research's concepts. The curves are highlighted in contrasting colors: blue for the curve obtaining through the proposed methodology and red for the curve obtained through the three-dimensional machine, as illustrated in Fig. 25.8.

The authors proposed to continue the work since substantial extra research needed before these concepts could become a practical reality, such as further investigation of image processing techniques for conversion to 3D models using CAD/CAM systems.

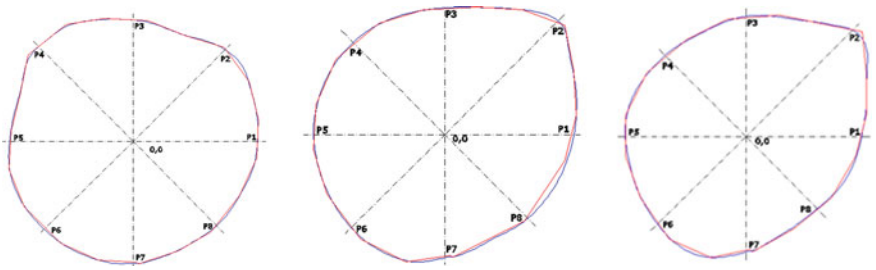


Fig. 25.8 Overlapped curves for the three sections respectively

25.5.2 Design and Development of a Conceptual System and Device for Acquisition, Conversion, Transmission and Processing of Biomedical Data

Biomedical data acquisition in real time has become more important in sports training and even in physical activities of everyday life as they can help the health or sport professionals supervise, monitoring and adjust the level and rhythm of the exercises. These data are usually acquired through specific equipment and are transferred manually to a computer to be analyzed allowing the adjustment in training to get the best performance. This system of data acquirement takes time, cost and can induce to errors since the data will be analyzed separately and not at the same time of the event/acquisition. In this way, there is a necessity for a system that is able to acquire the biomedical signals and send remotely to a computer where it will be analyzed in real time.

The integrated engineering system and a reduced size device developed by Boothroyd et al. [27] allows biomedical data acquisition operating in an integrated way: the biomedical signals are acquired by sensors and transmitted to a computer where the information is analyzed and converted into interpretative charts in real time.

The research is focused in three biomedical signals: oximetry, pedometrics and body temperature that can provide important information of how the body is responding to the training or exercises. Figure 25.9 shows the developed system for data acquisition, transmission and interpretation. The biomedical signals, in the pulse, oximeter were acquired by red and infrared light and converted into analogical and distinct electric impulses through two sensors/light receptors which were put in the index finger tip. The corporal temperature is easily obtained by the contact of the human skin with the sensor, put in the hand palm which uses the electric conductivity variation (ohmic resistance) to generate a distinct and analogical electric signal. The pedometry information was acquired by an electric switch placed inside the shoes, in contact with the sole of the foot and based in the electric switch on/off.

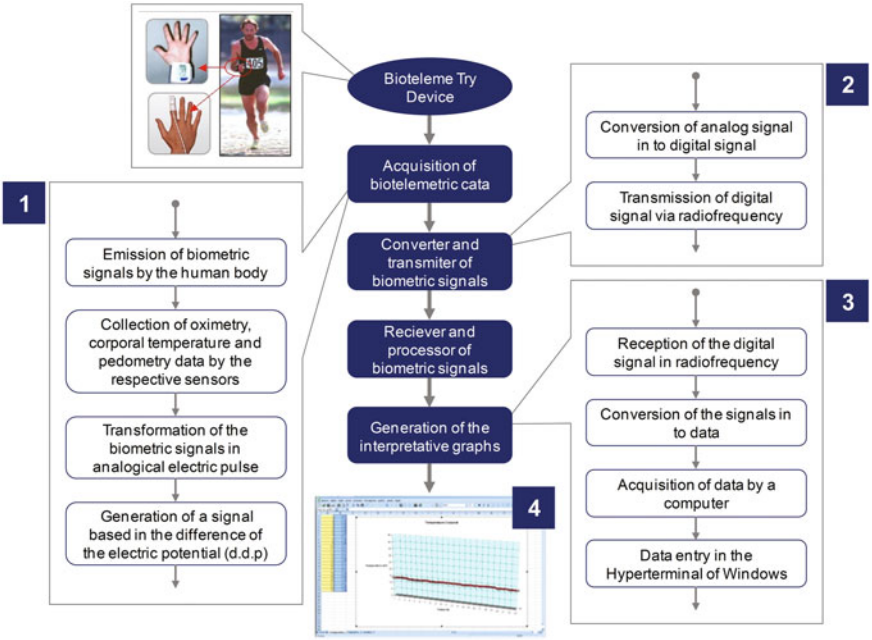


Fig. 25.9 Developed system for data acquisition, transmission and interpretation

The data obtained were analogical and needed to be converted into digital using a proper circuit (A/D) which was inside a micro controller for the radio frequency transmission (RF). As there were four distinct signals to be transmitted, they were serialized and transmitted orderly with milliseconds between them. The same process can be executed by a micro controller programmable with FLASH memory, through encoding lines. The signals enters the remote computer via serial door and can be observed, in real time, through the hyperterminal of Microsoft Windows which generates a document in text format with all the data. This document is loaded through a Macro developed by Microsoft Excel that generates a worksheet and from there, it was converted into interpretative charts. These charts can be divided in: Pulse oxymetry, showing the cardiac frequency and blood oxygenation rate in function of time; Pedometrics showing the distance and walking rhythm of the individual in function of time; and the Corporal temperature in function of time.

To validate the research, the system was applied to 5 volunteers. Tests were realized in a physiotherapy clinic with acclimatized environment using a motorized treadmill with a pre-adjusted distance of 50 m and medical supervision. From the data acquired it was possible to reach two distinct displays of interpretative charts: the first showing the relation between the pace length and the leg length,

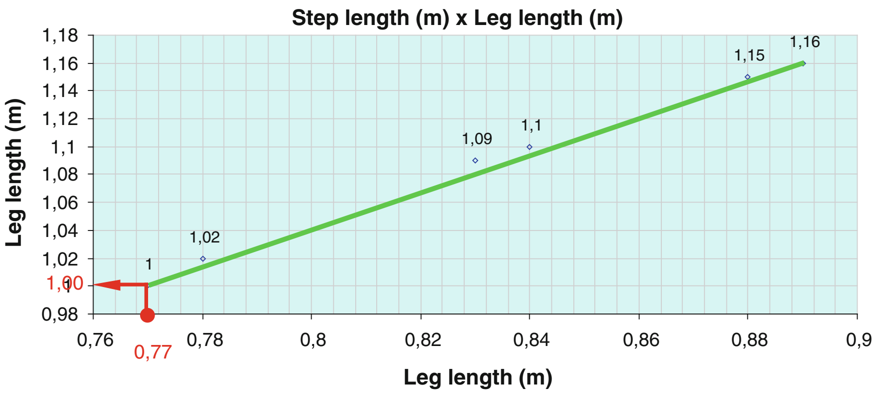


Fig. 25.10 Example of the relation between the pace length and the leg length

exemplified in Fig. 25.10; and the second showing the variation of temperature, the walked distance, number of paces, time between paces, and the oxymetry presented in Fig. 25.11.

The results showed that the developed system and device is able to obtain the information in a practical and reliable way, presenting normal standards values and not presenting significant errors. Therefore, the integrated engineering system

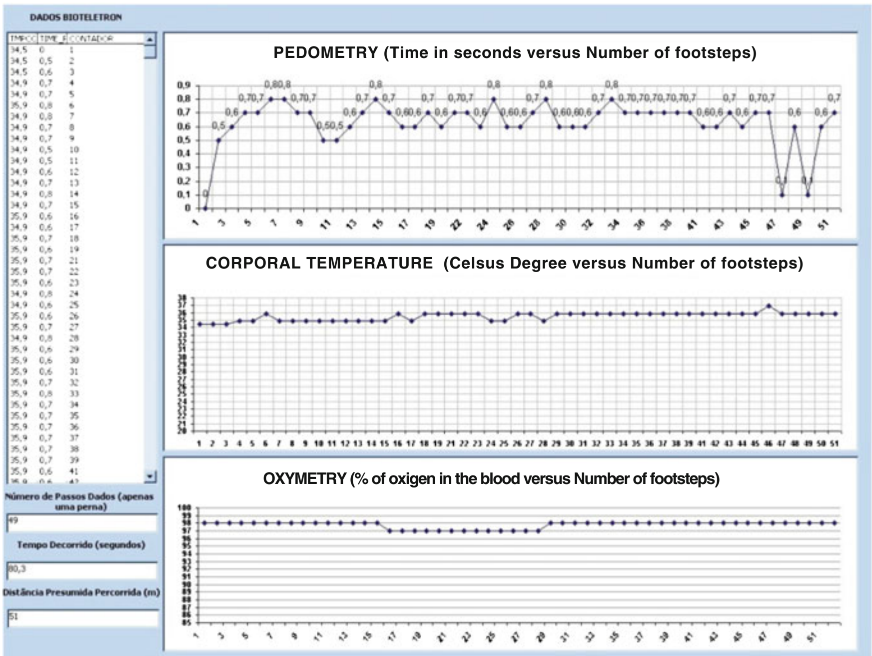


Fig. 25.11 Example of the displays presented by the device

developed to acquire, convert, transmit and process the biomedical data proved to be reliable and accurate and able to incorporate different concepts into a unique device. Further researches can be done as modifications and innovations can be implemented with the evolution of resources and tools, such as software and hardware, graphic interface or even the device miniaturization aiming the user's comfort.

25.5.3 Methodology to Determine a Suitable Implant for a Single Dental Failure

The multidisciplinary approach to achieve technological solutions in health care has improved surgical processes applying concepts of product engineering, especially in the medicine and dentistry areas. The use of CE concepts in dentistry improves the results since different points of view converge to a solution that is more reliable. Dental implant is a multivariable process that depends on the surgeon dentist expertise in most of the time. Some existing computer systems help with the visualization of CT images, but do not provide enough information for planning the dental implant process and do not support the process of selecting the implants that suits the patient best [50, 51]. In this context, a conceptual system is proposed based on techniques of medical and dental implant image processing which is able to provide support to the surgeon's decision making to define the single implant that best suits the patient [52, 53].

The dental implant selection is a simultaneous and interdependent analysis process that includes aspects as bone structure, nerves positioning, geometry of the mouth and teeth. For single implants placed between teeth, it is necessary to check the space available for inserting the implant. Figure 25.12 presents the conceptual methodological approach for determining the single dental implant, whose mark ("?") presents the necessary investigation to build an informational structure that provides support to the process.

The proposed system is divided in two parts:

(a) the product model where the requirements and specifications of all necessary information are defined to support the design system oriented on to single dental implant process; and
(b) the design system oriented to single dental implant process where the inference mechanisms for conversion, translation and sharing information between representations are defined.

The concept of inference mechanisms which are specialized systems elements capable of searching the necessary rules to evaluate and ordain logically the heuristic process of inference was used for the development of design system oriented to the process of single dental implant (DOSDI). The mechanisms for definition of the region of interest and geometric modeling of the symmetry axis and the

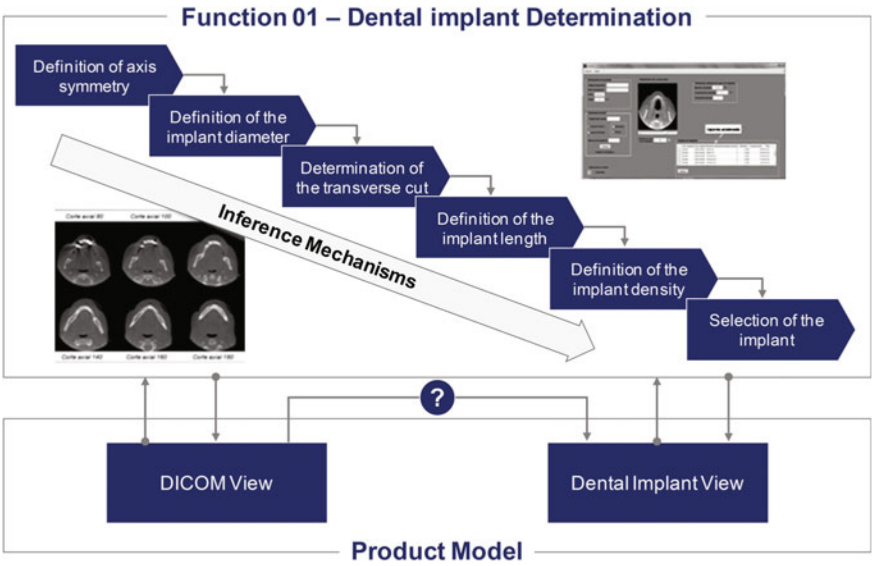


Fig. 25.12 Methodological proposal to determine a suitable implant for a single dental implant

mechanism of defining the implant diameter are responsible for the geometrical definition of the implant.

These mechanisms are intended to select the region of interest and the geometric modeling of the symmetry axis where the insertion of the dental implant occurs allowing the conversion of the axial cuts into transverse cut of this region. The definition of the region of interest is made by the oral facial dentist surgeon through observation/analysis, identifying the image that presents in detail the gaps between two teeth. The system intervenes and performs the geometric modeling of the dental arch from this image. From the region of interest is extracted only the bone information from processing these images using as segregation parameter, the Hounsfield Scale. As a result, the information of the bone geometry and the failure surrounding teeth is obtained, permitting a geometric analysis of the dental arch. Figure 25.13 illustrates a case study of a partial edentulous with a single failure in the canine region.

The geometrical analysis detects the geometrical center through the intersection of two reference lines that are based on the inner edge of the teeth failure neighbors (Fig. 25.14). This geometric center is used as a reference generating a symmetry line to the neighboring teeth which allows the extraction of the insertion center and the implant diameter as showed in Fig. 25.15.

The system delineates failure's axis of symmetry and together with the dentist it generates an accurate axis, relying only on images with an uncertainty grade of 0.25 mm, which is considered insignificant in the implantology area. The system defines the implant diameter, Table 25.1, based on the bone thickness obtained by

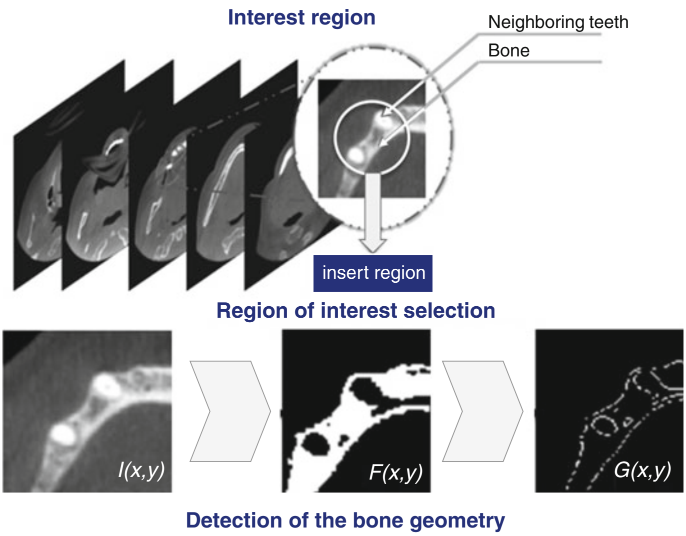


Fig. 25.13 Structure of the function of dental implant determination

Fig. 25.14 Symmetry axis construction in the dental arch

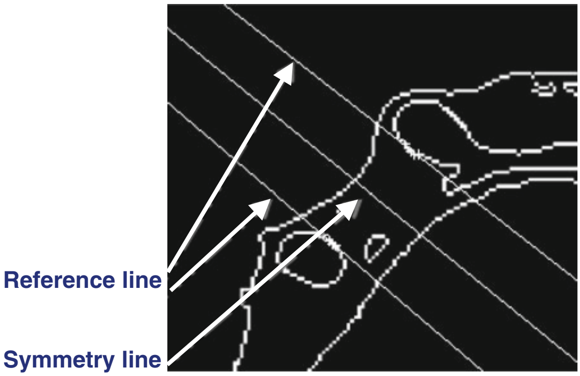


Fig. 25.15 Identification of the insertion point in the dental arch

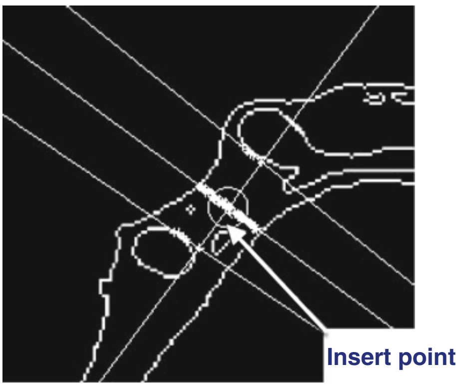


Table 25.1 Example of implant models obtained through the system

Implant body type	Implant body model	Bone density	Implant body diameter (mm)
CONE MORSE	TITA MAX CM	2	3.5
CONE MORSE	TITA MAX CM	2	3.5
CONE MORSE	TITA MAX CM	2	3.5
CONE MORSE	TITA MAX CM	2	3.75
CONE MORSE	TITA MAX CM	2	3.75
CONE MORSE	TITA MAX CM	2	3.75
CONE MORSE	TITA MAX CM	2	4.0
CONE MORSE	TITA MAX CM	2	4.0
CONE MORSE	TITA MAX CM	2	4.0
HEXAGONO INTERNO	TITA MAX IIPLUS	2	3.75
HEXAGONO INTERNO	TITA MAX IIPLUS	2	3.75
HEXAGONO INTERNO	TITA MAX IIPLUS	2	3.75

geometric modeling. For this specific case of study, it returned a selection of 12 models of dental implant that meet the design requirements delegating to the surgeon the identification of the most likely implant to be used. Since the number of implants offered by the manufacturer used in the research is approximately 150, the options were decreased by 92 % with the system turning the dental implant process more reliable.

The authors suggest for further researches the study of dental implant process in total edentulous with the construction of a guide mask and the identification of the insertion's depth as well as the drilling and threading rotation process.

25.6 Final Considerations

This study is a conceptual view on the CE philosophy importance in multidisciplinary engineering-medicine environment for IPDP of medical equipment. The level of complexity in the development of medical equipment requires the mastery of several technological areas of knowledge for the holistic design of the required product concept. The CE environment ensures the IPDP of products is oriented on health as it gives a wide view, understanding more clearly the planning aspects throughout the product life cycle.

To improve the innovation performance, it is necessary to develop a conceptual framework that is able to describe the IPDP process connecting the knowledge with the skills and competencies necessary to control and strategically manage the development cycle. So that, each design phase requires strategic planning to seek the best solutions once it should consider the specificity of users and medical equipment. Indeed, if during the phases of product development more information and experiences are introduced, formalized and systematized in the processes, higher will be the IPDP level of maturity, which will reflect in the quality of the final product.

It is worth noting that it is necessary further studies to develop models of tools specifically oriented to medical area, which would interact properly between the medicine-engineering fields absorbing the demands of the users and the market tendencies [15]. An additional important topic is the interface development that allows compatibility among tools and methods from different fields, so that, the equipment could fully attend the requirements of the IPDP involved areas [54].

CE environment gives a comprehensive view of IPDP and enables the interaction of distinct areas due to its multidisciplinary characteristic once it takes in consideration the users' demands and desires as well as the market tendencies for high technological equipment with lesser invasive procedures and better technical, clinical and economic quality [30, 55]. Therefore, CE in the IPDP of medical equipment contributes for medicine and engineering areas evolution since it conducts the technological and clinical innovation and brings new challenges in the searching for solutions to contemporary health issues [56].

Acknowledgments The authors are thankful for the financial support provided by the Coordination for the Improvement of Higher Level Personnel—CAPES and Pontifical Catholic University of Paraná—PUCPR.

References

1. ANVISA—Brazilian Health Surveillance Agency (in portuguese: Agência Nacional de Vigilância Sanitária), 2010. Resolução RDC n° 02, de 25 de Janeiro de 2010. Dispõe sobre o gerenciamento de tecnologias em saúde em estabelecimentos de saúde. Brasília. http://wwwbrasilsus.com.br/legislacoes/anvisa/102722-2.html. Accessed 15 Aug 2013
2. WHO, World Health Organization (2003) Medical device regulations: global overview and guiding principles. WHO, France
3. FDA, Food and Drug Administration (2013) Medical device: classify your medical device. http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice/. Accessed 10 Aug 2013
4. WHO, World Health Organization (2010) Medical devices: managing the mismatch: an outcome of the priority medical devices project. WHO Press, Geneva
5. Calil SJ, Teixeira MS (1998) Gerenciamento de manutenção de equipamentos hospitalares (Série Saúde & Cidadania, vol 11). São Paulo: Faculdade de Saúde Pública da Universidade de São Paulo
6. Olson WH (2009) Basic concepts of medical instrumentation. In: Webster JG (ed) Medical instrumentation: application and design, 4th edn. Wiley and Sons, NewYork-USA
7. Wang B (2012) Medical equipment maintenance: management and oversight. In: Enderle JD (ed), Synthesis lectures on biomedical engineering. University of Connecticut, Morgan & Claypool Publishers, San Rafael

8. Brazil (2013) Brazilian ministry of health guidelines methodological: preparation of studies for the evaluation of medical care equipment/Ministry of Health, Brazilian Department of Science, Technology and strategic inputs. Ministry of health (in portuguese: Ministério da Saúde. Diretrizes metodológicas: elaboração de estudos para avaliação de equipamentos médicos assistenciais/Ministério da Saúde, Secretaria de Ciência, Tecnologia e Insumos Estratégicos, Departamento de Ciência e Tecnologia. Ministério da Saúde), Brasília
9. HTAi Health Technology Assessment International, INAHTA International Network of Agencies for Health Technology Assessment (2013) Resources for health technology assessment—HTAi and INAHTA's white paper to WHO. http://www.inahta.org/upload/HTA_resources/AboutHTA_Resources_for_HTA.pdf. Accessed 10 Aug 2013
10. WHO World Health Organization (2011) Health technology assessment of medical devices. WHO Medical device technical series, Geneva
11. Back N, Ogliari A, Dias A, Silva JC (2008) In: Manole B (ed) Integrated product design: planning, design and modeling (in portuguese: Projeto integrado de produtos: planejamento, concepção e modelagem). São Paulo
12. Centro Cochrane do Brasil (2013). Saúde Baseada em Evidências. 2012. http://www.centrocochranedobrasil.org.br/cms/index.php?option=com_content&view=article&id=4&Itemid=13. Accessed 11 Aug 2013
13. Cochrane (2013) Evidence-based health care and systematic reviews. The Cochrane Collaboration. http://www.cochrane.org/about-us/evidence-based-health-care. Accessed 11 Aug 2013
14. Shavit O (2009) Utilization of health technologies: do not look where there is a light, shine your light where there is a need to look! Relating national health goals with resource allocation decision-making, illustration through examining the Israeli healthcare system. Health Policy
15. Medina LA, Okudan Kremer GE, Wysk RA (2013) Supporting medical device development: a standard product design process model. J Eng Des 24(2):83–119
16. Martina JL, Clark DJ, Morgan SP, Crowe JA, Murphy E (2012) A user-centred approach to requirements elicitation in medical device development: a case study from an industry perspective. Appl Ergon 43(1):184–190
17. Money AG, Barnett J, Kuljis J, Craven MP, Martin JL, Young T (2011) The role of the user within the medical device design and development process: medical device manufacturers' perspectives. BMC Med Inf Dec Making 11:15
18. Gad SC, Spainhour CB (2011) Medical device development. In: Gad SC, Spainhour CB (eds) Contract research and development organizations: their role in global product development. Springer, New York, pp 39–52
19. Sônego FS (2007) Estudo de métodos de avaliação de tecnologias em saúde aplicada a equipamentos eletromédicos. (Master Dissertation)—Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, Brazil
20. Kruglianska I (1992) Engenharia Concorrente: organização e implantação em empresas brasileiras. In: XVII Simpósio de gestão da inovação tecnológica—Anais, São Paulo
21. Cleetus KJ (1992) Definition of concurrent engineering. Concurrent Engineering Research Center—Technical report. West Virginia University, Morgantown
22. Hunt VD (1993) Reengineering: leveraging the power of integrated product development. Essex Junction, Oliver Wight
23. Hartley JR (1998) Simultaneous engineering (in portuguese: Engenharia, simultânea edn. Bookman, Porto Alegre, Brazil
24. Rozenfeld H, Forcellini FA, Amaral DC, Toledo JC, Silva SL, Alliprandini DH, Scalice RK (2006). In: Saraiva (ed) Management of product development—a reference to process improvement (in portuguese: Gestão de desenvolvimento de produtos—Uma referência para a melhoria do processo). São Paulo
25. Ito T (2012) A concurrent engineering approach towards digital dentistry support. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering, 2013, Springer, London, pp 231–242

26. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manag 7(2):101–115
27. Boothroyd G, Dewhurst P, Knight WA (2011) Product design for manufacture and assembly, 3rd edn. CRC Press, New York
28. Hauser JR, Clausing D (1988) The house of quality. Harv Bus Rev 63–73
29. Prasad B (1996) Concurrent engineering fundamentals: integrated product and process organization. Prentice Hall, New Jersey
30. Medina LA, Wysk RA, Okudan Kremer GE (2011) A review of design for X methods for medical devices: the introduction of a design for FDA approach. In: ASME 2011 international design engineering technical conferences and computers and information in engineering conference. vol 9, 23rd International conference on design theory and methodology; 16th design for manufacturing and the life cycle conference. ASME, Washington 28–31 Aug 2011, pp 849–861
31. Kretschmer R, Rulhoff S, Stjepandić J (2014) Design for assembly in series production by using data mining methods. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering, IOS Press, Amsterdam, pp 379–388
32. Terziovski M, Morgan JP (2006) Management practices and strategies to accelerate the innovation cycle in the biotechnology industry. Technovation 26:545–552
33. Chang D, Chen CH (2014) Understanding the influence of customers on product innovation. Int J Agile Syst Manag 7(3–4):348–364
34. Okumura MLM (2012) Concurrent engineering applied to the design of integrated development of inclusive products: a proposed conceptual framework (in portuguese: A engenharia simultânea aplicada ao projeto de desenvolvimento integrado de produtos inclusivos: uma proposta de framework conceitual). Master Dissertation, PPGEPS–PUCPR, Curitiba, Brazil
35. Lee S, Yang J, Han J (2012) Development of a decision making system for selection of dental implant abutments based on the fuzzy cognitive map. Expert Syst Appl 39(2012):11564–11575
36. Park SG, Lee S, Kim MK, Kim HG (2012) Shared decision support system on dental restoration. Expert Syst Appl 39(2012):11775–11781
37. Iida I (2005) Ergonomia: projeto e produção. 2 edn revi. e ampli. São Paulo, Edgard Blücher
38. Buzzi M, Colombo G, Facoetti G, Gabbiadini S, Rizzi C (2012) 3D modelling and knowledge: tools to automate prosthesis development process. Int J Interact Des Manuf 6:41–53
39. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3–4):324–347
40. Trappey CV, Wang TM, Hoang S, Trappey AJC (2013) Constructing a dental implant ontology for domain specific clustering and life span analysis. Adv Eng Inf 27(2013):346–357
41. Nicholds BA, Mo J (2014) Determining an action plan for manufacturing system improvement: the theory. Int J Agile Syst Manag 6(4):324–344
42. Okumura MLM, Canciglieri O Jr (2013) O Desenvolvimento de Produtos envolvendo a Tecnologia Assistiva por meio de Estudo de Caso Múltiplo. 9° Congresso Brasileiro de Gestão e Desenvolvimento de Produtos, Universidade Federal do Rio Grande do Norte, UFRN—Natal (in portuguese)
43. Wiklunk ME (2005) The rising bar, medical product design excellence. In: Wiklund ME, Stephen B, Wilcox SB (eds) Designing usability into medical products. CRC Press, Florida
44. Greboge T, Rudek M, Jahnen A, Canciglieri O Jr (2013) Improved engineering design strategy applied to prosthesis modelling. In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 60–71
45. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manag 7(2):116–131
46. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Syst Manag 6(1):7–24
47. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manag 6(4):307–323

48. Canciglieri O, Rudek M, Francesconi T, Souza TM (2010) Geometric modeling from the medical images in the CAD system to support prosthesis design. In: Proceedings of the 40th international conference on computers and industrial engineering—CIE40, CIE—CD-Rom, 2010, Awaji Island—Japan. IEEE Kansai Section: IEEE, 2010
49. Canciglieri O Jr, Rudek M, Greboge T (2011) A prosthesis design based on genetic algorithms in the concurrent engineering context. In: Frey DD et al (ed) Improving complex systems today, Proceedings of the 18th ISPE international conference on concurrent engineering. Springer, London, pp 12–24
50. Chang M, Park SC (2009) Automated scanning of dental impressions. Comput Aided Des 41 (2009):404–411
51. Fang JJ, Kuo TH (2009) Tracked motion-based dental occlusion surface estimation for crown restoration. Comput Aided Des 41(2009):315–323
52. Szejka AL, Rudek M, Canciglieri O Jr (2012) A reasoning system to support the dental implant planning process. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 909–919
53. Szejka AL, Pereira JA, Rudek M, Canciglieri O Jr (2013) Methodological proposal to determine a suitable implant for a single dental failure through cad geometric modelling. In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 303–313
54. Bondar S, Ruppert C, Stjepandić J (2014) Ensuring data quality beyond change management in virtual enterprise. Int J Agile Syst Manag 7(3–4):304–323
55. Januszka M, Moczulski W (2012) Acquisition and knowledge representation in the product development process with the use of augmented reality. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment, Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 219–229
56. Ogrodnik PJ (2013) Medical device design: innovation from concept to market. Elsevier, Oxford

Chapter 26
Carbon Emission Analysis for Renewable Energy Policies

Amy J.C. Trappey, Charles V. Trappey, Jerry J.R. Ou, C.T. Hsiao, Kevin W.P. Chen and Penny H.Y. Liu

Abstract Countries and government regions are promoting renewable energy to effectively reduce carbon emissions. However, the carbon footprint of a given industry in a specific region is hard to measure and the long-term effect of an untested green policy for carbon reduction is difficult to predict. This chapter introduces an approach that combines economic input-output life cycle assessment (EIO-LCA) and a location quotient (LQ) to measure regional carbon footprints using local environmental and industrial data. The results enable government policy makers to accurately formulate policies that target critical contributors while simulating the economic impact using system dynamics (SD) modeling. In the case study, policy scenarios are simulated to evaluate the time-varying impacts of proposed green transportation strategies for Taiwan's low carbon island (Penghu Island) pilot project. The methodology provides a generalized tool for green energy policy assessment. This chapter is the extension of the original research reported by the authors in Trappey et al. (Energy Policy 45:510-515 [1], Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Springer, London, pp. 367–377 [2]).

A.J.C. Trappey (✉) · P.H.Y. Liu
Department of Industrial Engineering and Engineering Management,
National Tsing Hua University, Hsinchu 30013, Taiwan
e-mail: trappey@ie.nthu.edu.tw

C.V. Trappey
Department of Management Science, National Chiao Tung University, Hsinchu, Taiwan

J.J.R. Ou
Department of Business Administration, Southern Taiwan University, Tainan, Taiwan

C.T. Hsiao
Department of Economics, Tunghai University, Taichung, Taiwan

K.W.P. Chen
Department of Industrial Engineering and Management,
National Taipei University of Technology, Taipei, Taiwan

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_26

Keywords Low carbon island · Green transportation · Economic input–output life cycle assessment · Carbon footprint · System dynamics

26.1 Introduction

Global warming has caused international communities to place greater emphasis on environmental protection. Carbon dioxide (CO_2) emissions are a primary factor causing the greenhouse effect. According to the report of Intergovernmental panel on climate change (IPCC) [3], the average temperature on earth is estimated to increase 4 °C and sea levels will rise about 0.5 m by 2100, if global warming problems are not addressed. Therefore, to control and decrease CO_2 emissions are critical issues. Based on the statistical report published by International energy agency (IEA) in 2006, many developed and developing countries, with manufacturing based economies, generate much higher CO_2 emissions compared to other countries. For example, the population of Taiwan accounts for 0.35 % of the world population, but the proportion of CO_2 emissions is equal to 0.7 % of the world's emissions [1, 2, 4]. Thus, the Taiwan government passed the Sustainable Energy Policy on World Environment Day in 2008 and committed to improve energy efficiency, develop clean energy, and ensure a stable energy supply. To develop clean energy, Taiwan implemented a feed-tariff mechanism as the incentives policy [4]. Taiwan also initiated a Master Plan of Energy Conservation and Carbon Mitigation in 2009 and announced that 2010 was the start year for promoting energy savings and carbon reduction.

One part of the Master Plan of Energy Conservation and Carbon Mitigation is called the Penghu low carbon island development project. Under this main project, seven specific low carbon operational sub-projects and measures were chosen as pilot runs covering spectrums of renewable energy (RE), energy conservation, green transportation, low carbon construction, environmental greening, resource recycling, and low carbon living [5–7]. The project applies energy conservation and carbon mitigation technology to build a clean low carbon island and benchmarks these pilot experimental results as a means to transform all cities and environmental practices in Taiwan. Determining whether the policy implementation effect will achieve the objectives of conserving energy and reducing carbon emissions have become a focus point of policy makers, academics, and the full range of stakeholders.

Building a low carbon island is a complicated system engineering challenge, since it affects factors related to environment, economy and society as a complicated and intertwined ecological, economic, and social system. It also requires a concurrent engineering (CE) approach to maximize positive outcomes and minimize negative impacts. On the whole, the population, the ground forest area, industry and commercial activities, transportation, people's daily energy usage, and CO_2 generation are only a few of the critical issues to consider. Even on a small island where many variables can be easily controlled, these factors interact with each other and, thereby, form a complicated and dynamic system reflecting the causal relationship of the factors in regard to the energy consumption and carbon emission issues. While there are many sub-projects included in Penghu Low Carbon Island Development Project, the demonstration of the policy's efficacy is a risky venture unless the results are valid and reliable. Trappey et al. [8] indicate that each country's situation is unique and the problem of implementing low carbon emission policies must be dissected into logical parts (or geographic and industrial clusters) for analysis, pilot plant experimentation, and then gradual implementation. Thus, this research develops a stepwise evaluation approach of the green transportation strategy to support low carbon policy development on Penghu Island to demonstrate the proposed methodology. First, we use economic input–output (IO) life cycle assessment (EIO-LCA) and the location quotient (LQ) method to estimate the carbon footprint (CF) of the Penghu County industrial sector. Afterwards, using the results of the EIO-LCA method, this study constructs a causal feedback loop diagram to represent how the green transportation policies decrease CO_2 emissions. Finally, a system dynamics (SD) model simulates the scenarios and evaluates the time-varying impacts of the proposed carbon reduction strategies.

26.2 Literature Review on Carbon Reduction Policies and Research

The related literature reviews introduce CF, low carbon island, and applications of EIO-LCA and SD approach.

26.2.1 Carbon Footprint

The Carbon Trust [9] says that a carbon CF measures the total greenhouse gas emissions caused directly and indirectly by a person, an organization, an event or a product. Minx et al. [10] also define CF as the direct and indirect greenhouse gas emissions and is measured in tons of CO_2 equivalent. The CF is used in many applications analyzing national emission inventories and trade, supply chains, organizations, consumption patterns and lifestyles, and regional and local CF [10].

The IO analysis is used to calculate the CFs for small spatial areas and typically for municipalities and cities. However, the main challenge associated with estimating local CFs based on IO analysis is combining information on global production activities with information on local consumption activities. As the importance of local mitigation and adaptation measures are increasingly recognized, there is greater application in development of policies [10].

26.2.2 Low Carbon Island

The low Carbon Island is also known as the RE Island and the Sustainable Energy Island. Chen et al. [11] said that island development problems are mostly related to imported fossil fuel energy dependence, fresh water availability, waste management, and transportation problems. However RE technology is one solution which produces energy by transforming natural phenomena (or natural resources) into useful energy forms.

There are many internationally successful and renowned RE island examples, such as Greece Dodecanese Island [12], Gökceada Island, Turkey [13], Yakushima Island, Japan [14], and Denmark's Samso Island. After 10 years' work for the development of RE, Denmark's Samso Island reach many objectives including self-sufficiency, exploitation of local resources, the supply and utilization of heat, electricity production, and improvement of economy, employment, and environment [15].

26.2.3 EIO-LCA

EIO-LCA is used to assess the use of raw materials, energy resources, and environmental emissions in economic activities [16]. Compared to the process-based LCA, the EIO-LCA not only improves some process bottlenecks and expands the system scope to include an entire economy of a country or region [17] with increased reliability [18]. The EIO-LCA was proposed by Leontief in 1936, and expresses the economic relationship of each sector of a given country or region using equilibrium theory. In recent years, IO analysis has been used to assess the energy consumption and environmental impact of merchandise, service, countries, and regions [10].

General economic models are based on assumptions and EIO-LCA is based on the follows three basic assumptions [19]:

1. Each industry produces only single product.
2. There is a linear relationship between each industry output and a fixed input ratio.
3. Every production of each industry has constant return to scale.

In order to utilize the IO approach to conduct LCA, the EIO-LCA uses data derived from the bureau of economic analysis (BEA) and publicly available environmental data from the U.S. Environmental Protection Agency (EPA) and the Department of Energy (DOE) [20]. In Taiwan, EIO-LCA uses the data derived from the report of industrial statistics published by the Directorate-General of Budget, Accounting and Statistics of the central government.

Recently, EIO-LCA methods have been applied to calculate CO_2 and GHG emissions [21–24]. Acquaye and Duffy [25] used the EIO-LCA method to estimate the energy and GHG emissions intensity of the Irish construction sector and estimated its contribution to overall Irish national emissions. Trappey et al. [26] applied EIO-LCA method to evaluate the carbon emissions of new products and identified problematic carbon emissions in the supply chain. Ju and Chen [27] also used the EIO-LCA method to evaluate the upstream CF and CO_2 emissions of 28 economic sectors of Chongqing area in China and identified the significant sectors that contribute the most to climate change. Further, this research uses EIO-LCA to effectively assess the industrial sectors' CF in Taiwan's Penghu County. According to the results of the carbon emission of the local industrial sectors, those are not only a benchmark for reference but also enable the government to plan carbon reduction policies.

26.2.4 System Dynamics

SD was proposed by Forrester in 1956. SD is a modeling approach to describe the activities of complex systems over time incorporating the control factors of systems while observing plausible reactions and behaviors. Therefore, SD is commonly applied to assess the mid- to long-term effects of decisions or policies when the causal factors of systems are complex and dynamic [28]. Thus, SD analysis can be viewed as a CE approach that simultaneously considers the impact of policies prior to the actual implementation.

Several SD research studies have been used to evaluate environmental-related public policies. Wang et al. [29] presented a SD approach based on the cause-and-effect analysis and feedback loop structures in urban transportation systems. Jin et al. [30] developed a dynamic ecological footprint forecasting platform to support policy making for urban sustainability improvement. Han and Hayashi [31] studied the inter-city passenger transport in China as a case and developed a SD model for policy assessment and CO_2 mitigation potential analysis. Trappey et al. [5] applied the SD approach to analyze the solar energy policy implemented in Taiwan Penghu Island. According to the above SD literature review, many SD methods have been applied to access environmental impact and policies. Therefore, this paper uses EIO-LCA to evaluate the Penghu low carbon island development and applies the results to evaluate CO_2 emissions reduction by SD.

26.3 Methodology

The research process of carbon emission analysis is divided into three parts as shown in Fig. 26.1. First, the study uses the IO analysis and LQ method to assess the carbon emissions of industrial sectors for a given region before the implementation of carbon reduction policies. Secondly, the SD approach is applied to

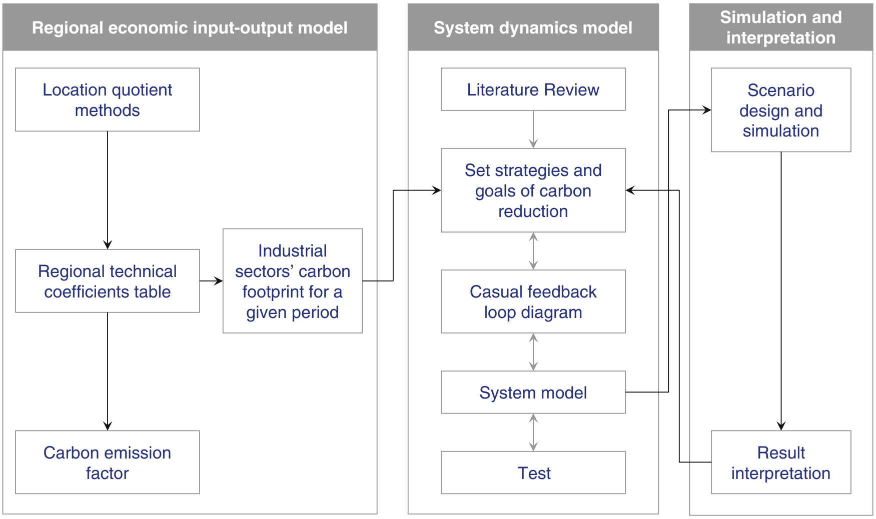


Fig. 26.1 Methodology for regional carbon emission analysis

construct the mathematical and quantitative model with causal feedback relationships based on the proposed carbon reduction policies in the region. Finally, different scenarios of the model are simulated and evaluated.

26.3.1 Regional Economic Input–Output Model

There are four steps to construct the EIO model taking into account the deviations of regional industries using the LQ adjustment.

Step 1: Construct the national IO transaction table. In Table 26.1, the intermediate output (O) and intermediate input (I) represent the flows of sales and purchases between sub-sectors. X_{ij} indicates the output of sub-sector i to the sub-sector j. The total input is the sum of the Intermediate input (I) and the added value (V). And, the total output is the sum of the Intermediate output (O) and the final demand (F) or GDP.

Step 2: After constructing the IO transactions table, the technical coefficient matrix A is derived from Table 26.1 using the following formula:

$$A = \begin{bmatrix} X_{11}/X_1 = a_{11} & X_{12}/X_2 = a_{12} & \ldots & X_{1n}/X_n = a_{1n} \\ X_{21}/X_1 = a_{21} & X_{22}/X_2 = a_{22} & \ldots & X_{2n}/X_n = a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1}/X_1 = a_{n1} & X_{n2}/X_2 = a_{n2} & \ldots & X_{nn}/X_n = a_{nn} \end{bmatrix}$$

Table 26.1 Input–output transactions table [16]

	Input to sectors				Intermediate output O	Final demand F	Total output X
Output from sectors	1	2	3	n			
1	X_{11}	X_{12}	X_{13}	X_{13}	O_1	F_1	X_1
2	X_{21}	X_{22}	X_{23}	X_{23}	O_2	F_2	X_2
3	X_{31}	X_{32}	X_{33}	X_{33}	O_3	F_3	X_3
4	X_{n1}	X_{n2}	X_{n3}	X_{nn}	O_n	F_n	X_n
Intermediate input I	I_1	I_2	I_3	I_n			
Value added V	V_1	V_2	V_3	V_n		GDP	
Total input X	X_1	X_2	X_3	X_n			

A represents the national technical coefficient matrix and the element a_{ij} in A represents the input of sub-sector i when sub-sector j is a unit output of product or service.

Step 3: Use the LQ method to effectively assess the regional technical coefficient matrix A^r. The LQ analysis is used to compare levels of employment between two geographic areas. The location quotient value (LQ_i) is expressed as:

$$\text{LQ}_i = \frac{E_{ir}/E_r}{E_{in}/E_n}$$

where E_{ir} and E_{in} are the employment population of the ith industry in a given region or nation whereas E_r and E_n are the total employment population in a given region or nation. If LQ_i is greater than one ($\text{LQ}_i > 1$), then the sector i in the region has a higher employment rate than the nation. If LQ_i is less than one ($\text{LQ}_i < 1$), it is assumed that the sector is not able to satisfy regional demand with its employment output. Therefore, the national coefficient of the ith sector can be adjusted by multiplying LQ_i to adjust the other regional coefficients. The formulas are shown as below [32]:

$$a_{ij}^r = a_{ij}, \quad \text{if } \text{LQ}_i > 1$$
$$a_{ij}^r = a_{ij} \times \text{LQ}_i, \quad \text{if } \text{LQ}_i < 1$$

where a_{ij}^r is an element of the regional technical coefficient matrix A^r. The regional technical coefficient matrix A^r is shown below.

$$A^r = \begin{bmatrix} a_{11}^r & a_{12}^r & \cdots & a_{1n}^r \\ a_{21}^r & a_{22}^r & \cdots & a_{2n}^r \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1}^r & a_{n2}^r & \cdots & a_{nn}^r \end{bmatrix}$$

Step 4: Use the Leontief Inverse matrix $(I - A^r)^{-1}$ and the following formulas are used to assess the quantity of CO_2 emissions in a given region:

$$X = (I - A^r)^{-1}F$$

$$B = RX$$

where X is the direct vector of required inputs, F is the vector of the desired output, I is the identity matrix, B is the vector of the environmental burden, and R is a matrix with diagonal elements representing the carbon emissions per unit (i.e., the coefficients of carbon emissions) provided by each sector. The carbon emission coefficients for all industrial sectors (R) are obtained from the government's Energy Bureau at the Ministry of Economic Affairs, the IPCC and other relevant secondary sources.

26.3.2 System Dynamics Model

In order to understand CO_2 emissions in industrial sectors of a given region and the corresponding benefits from the carbon reduction policies, this research constructs a causal feedback loop diagram based on SD. First, it is necessary to identify the problem and the boundary of the system. Then, the influencing factors are defined and cause-and-effect analysis is used to describe the system. Based on the results of cause-and-effect analysis, the quantitative system structure is constructed using the resulting coefficients and equations. After constructing the quantitative model, different scenarios are simulated and results are compared. Finally, the model is modified and tested to summarize the relevant conclusions and policy suggestions.

26.4 Case Study: Penghu Low Carbon Island Implementation

This research uses the low carbon island development project implemented on Penghu Island as the case study to demonstrate the proposed methodology. Penghu is the only government county of Taiwan which is an island located between Mainland China and Taiwan [33]. In order to promote local economic growth, Penghu is being developed as a natural resort and an environmental friendly island. The carbon island development project combines local features with different low carbon actions and applications to create a better environment. The IO analysis and LQ methods are used to assess the industrial sectors' CF. And then the study analyzes the green transportation policy and evaluates its benefits toward CO_2 emissions reduction [1, 2].

26.4.1 EIO-LCA for the Case

The IO model includes an IO table, a technical coefficients table, and a Leontief inverse matrix as described in Sect. 26.3.1. The case uses the input coefficients table of producers' prices [34] shown in matrix A. The LQ method is then used to estimate Penghu's technical coefficients matrix A^r. Finally, Penghu's Leontief inverse matrix $(I - A^r)^{-1}$ is calculated.

Based on the input coefficients table of producers' prices, 52 sub-industrial sectors are combined into 9 main sectors including Agricultural, Forestry, Fishery and Animal Husbandry (A), Manufacturing (B), Electricity, Gas and Water (C), Construction (D), Trade, Accommodation, and Food and Beverage (E), Transportation and Communication (F), Banking, Insurance Entities and Real Estate (G), Industry, Commerce and Services (H), and Social and Personal Services (I). The EIO technical coefficients for these nine sectors are shown in Table 26.2.

As listed in Table 26.3, the LQ of industries for Penghu Island are calculated based on the manpower survey statistics of 2006 [35], statistics of the agriculture, forestry, fishery and husbandry census from 2005 [36], and the industry, commerce and service census of 2006 [37]. Only the LQ for the Agriculture, Forestry, Fishery and Husbandry sector is greater than one ($LQ_A > 1$) which implies the employment for this island sector is greater on average than Taiwan island. The LQ of other sectors that are less than one (LQ_B, LQ_C, ..., $LQ_I < 1$) indicate that these sectors cannot satisfy Penghu's own economic demands. Thus, to adjust the coefficient of

Table 26.2 National technical coefficients for nine main sectors in 2006

Sector	A	B	C	D	E	F	G	H	I
A	0.4996	0.6567	0.0001	0.0023	0.0335	0.0000	0.0000	0.0013	0.0008
B	0.9719	13.8203	0.8287	0.5036	0.2584	0.5361	0.0173	0.3532	0.4806
C	0.0367	0.6340	0.5687	0.0026	0.0563	0.0278	0.0063	0.0412	0.0922
D	0.0100	0.0779	0.0394	0.0009	0.0111	0.0486	0.0585	0.0110	0.0251
E	0.3175	2.2313	0.1076	0.0924	0.1311	0.2511	0.0138	0.1082	0.1805
F	0.0518	0.4106	0.0440	0.0279	0.0578	0.6799	0.0306	0.1662	0.1365
G	0.0563	0.4367	0.0552	0.0111	0.1095	0.0755	0.2737	0.0951	0.1472
H	0.0468	0.8812	0.0945	0.0508	0.0923	0.2637	0.0911	0.1369	0.1179
I	0.0151	0.1023	0.0202	0.0091	0.0188	0.0629	0.0124	0.0465	0.0564

Table 26.3 Industrial sectors' location quotients for Penghu Island

Sector	Penghu, 2006		Nation, 2006		Location quotient
	Employment	Percentage	Employment	Percentage	
A	41,379	0.73	3,492,237	0.32	2.29 > 1
B	589	0.01	2,694,303	0.25	0.04 < 1
C	36	0.0006	59,777	0.01	0.11 < 1
D	1,629	0.03	477,444	0.04	0.69 < 1
E	8,623	0.15	2,209,752	0.20	0.74 < 1
F	1,077	0.02	538,738	0.05	0.41 < 1
G	350	0.006	376,602	0.03	0.17 < 1
H	373	0.007	441,808	0.04	0.17 < 1
I	2,434	0.04	647,410	0.06	0.68 < 1
Total	56,490	100	10,938,071	100	

Table 26.4 Penghu's total Leontief inverse matrix $(I{-}A^r)^{-1}$

Sector	A	B	C	D	E	F	G	H	I
A	2.4143	4.2579	0.1738	0.1053	0.1489	0.1842	0.0143	0.0957	0.1246
B	0.2570	2.8942	0.1127	0.0646	0.0491	0.1104	0.0085	0.0575	0.0751
C	0.0391	0.2907	1.0793	0.0076	0.0133	0.0190	0.0021	0.0124	0.0206
D	0.0517	0.3140	0.0444	1.0095	0.0178	0.0651	0.0447	0.0205	0.0328
E	1.1636	6.8117	0.3761	0.2374	1.2588	0.5778	0.0437	0.2595	0.3604
F	0.1865	1.1404	0.0793	0.0464	0.0636	1.4550	0.0258	0.1285	0.1215
G	0.0734	0.4424	0.0314	0.0143	0.0322	0.0455	1.0516	0.0314	0.0446
H	0.0901	0.6637	0.0478	0.0264	0.0333	0.0979	0.0207	1.0454	0.0470
I	0.0726	0.4279	0.0361	0.0186	0.0264	0.0866	0.0127	0.0493	1.0593

Penghu, we multiply the national coefficients by LQ_B, LQ_C, ..., LQ_I. After acquiring the technical coefficient table of Penghu, the Leontief inverse matrix $(I - A^r)^{-1}$ is calculated as shown in Table 26.4.

In addition to Penghu's Leontief inverse matrix $(I - A^r)^{-1}$, the final demand vector for tourism F is estimated for Penghu. In this paper, we estimate the final demand vector of tourism F (Table 26.5) based on the statistical reports of the number of visitors to principal scenic spots in Taiwan in 2006 and the travel time data of Taiwan residents in 2007. Afterward, the Leontief inverse matrix $(I - A^r)^{-1}$, the final demand vector of tourism F, and the formula $X = (I - A^r)^{-1}F$ are used to estimate the direct vector (X) of required economic inputs for all sectors are computed as shown in Table 26.6.

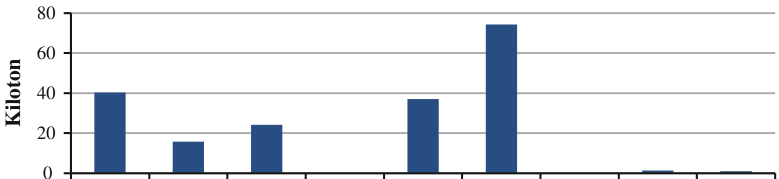
Finally, with Penghu's required inputs (X) for all industrial sectors, this research uses Lin and Huang [38] Taiwan resource utilization analysis and CO_2 emissions factors to derive CO_2 emission quantities by industrial sectors. Figure 26.2 depicts the CF assessment results using the LQ-based EIO-LCA method. The data show that the Transportation and Communication sector F (74 kt), Agricultural, Forestry,

Table 26.5 Final demand vector *F*

Sector	Final demand (million NTD)
Agricultural, forestry, fishery and animal husbandry	0
Manufacturing	0
Electricity, gas and water	0
Construction	0
Trade, accommodation and food and beverage	7,869.7
Transportation and communication	2,360.9
Banking, insurance entities and real estate	0
Industry, commerce and services	674.5
Social and personal services	337.3

Table 26.6 Direct vector of required inputs *X*

Sector	Required inputs (million NTD)
Agricultural, forestry, fishery and animal husbandry	1,713.1
Manufacturing	711.2
Electricity, gas and water	165.0
Construction	318.9
Trade, accommodation and food and beverage	11,567.5
Transportation and communication	4,062.8
Banking, insurance entities and real estate	396.9
Industry, commerce and services	1,214.3
Social and personal services	802.7



	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)
CO2 Emissions	40.26	15.79	24.12	0.255	37.02	74.35	0.079	1.336	0.963

Fig. 26.2 The carbon footprint assessment of industrial sectors by EIO-LCA. *A* agricultural, forestry, fishery and animal husbandry, *B* manufacturing, *C* electricity, gas and water, *D* construction, *E* trade, accommodation and food and beverage, *F* transportation and communication, *G* banking, insurance entities and real estate, *H* industry, commerce and services, *I* social and personal services

Fishery and Animal Husbandry sector A (40 kt), Trade, Accommodation, Food, and Beverage sector E (37 kt), and Electricity, Gas, and Water sector C (24 kt), and the Manufacturing sector B (16 kt) emit higher amounts of CO_2 than other sectors in Penghu.

26.4.2 System Model Construction of Case

The EIO-LCA method is used to assess the industrial sectors' carbon emission produced by past tourism development in Penghu County. The result shows that Transportation and Communication has the greatest impact on Penghu's environment. Therefore, the SD model is constructed with specific causal feedback loops to analyze the effectiveness of required investment cost and the corresponding benefits for green transportation development. And green transportation developments consist of following policies [7]:

1. Replace all gasoline motorcycles with electric scooters incrementally in year 2011, 2012 and 2013.
2. Limiting road use of two-stroke gasoline motorcycles.
3. Limiting road use of gasoline motorcycles licenses.

According to all of the above policies, we discuss operation process for total motorcycle levels, total electric scooters, total CO_2 emissions and others external factors to construct specific causal feedback loops (Fig. 26.3).

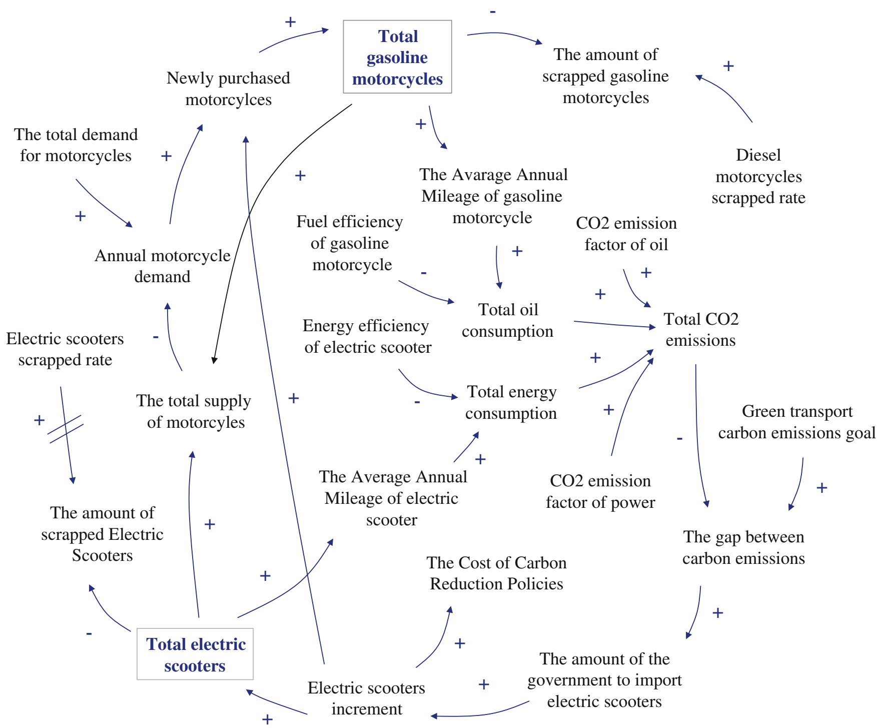


Fig. 26.3 Causal feedback loop for green transportation development

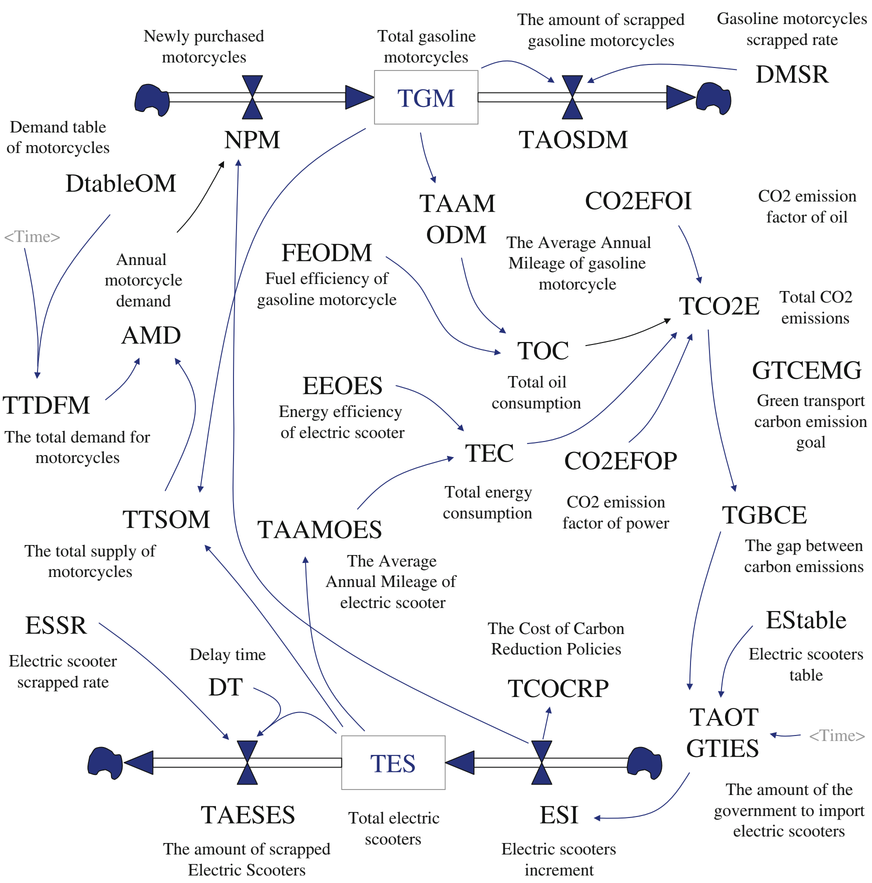


Fig. 26.4 The quantitative SD model for the green transportation pilot project

Afterward, the mathematical model with causal feedback relationships is constructed based on the proposed green transportation strategy (Fig. 26.4). The variables are identified and relevant equations are established based on the feedbacks and cause-and-effect loops. Next, we introduce sub-model of the green transportation policy.

1. Total Electric Scooters Sub-model (TES)
   Total electric scooters sub-model reflects the amount of electric scooters that the government imported. If carbon emission gap between total CO_2 emissions and green transport carbon emission goal is positive, then the government imports electric scooters continually. The government stops the import policy if the carbon emission gap is zero or negative.

Table 26.7 Parameter values for the SD model of the Penghu Low Carbon Island [2]

Parameters	Value
CO_2 emission factor of oil	2.263 kg-CO_2/L
CO_2 emission factor of power	0.623 kg-CO_2/W
Fuel efficiency of gasoline motorcycles	40 km/L
Energy efficiency of electric scooters	20 km/W
Gasoline motorcycles scrap rate	1.4605 %
The average annual mileage of gasoline motorcycles	4,099 km/unit
The average annual mileage of electric scooters	3,000 km/unit
The investment cost of electric scooters	1,000 US$/unit

2. Total CO_2 Emissions Sub-model (TCO_2E)
   Total CO_2 emissions sub-model reflects the total CO_2 emissions of gasoline motorcycles and electric scooters. The formulas for total CO_2 emissions of gasoline motorcycles and electric scooters are shown as follows:
   (a) Total oil consumption = The average annual mileage of gasoline motorcycles/Fuel efficiency of gasoline motorcycles
   (b) Total energy consumption = The average annual mileage of electric scooters/Energy efficiency of electric scooters
   (c) Total CO_2 emissions of gasoline motorcycles and electric scooters = Σ(Fuel consumption $\times$ CO_2 emission factor of fuel)
3. Total Gasoline Motorcycles Sub-model (TGM)
   Total gasoline motorcycles (TGM) sub-model reflects the demands for buying gasoline motorcycles. The newly purchased motorcycles value is the maximum value which between annual motorcycles of demand deducts electric scooters increment and zero. And the annual motorcycle demand value which equals to the total demand for motorcycles deducts the total supply motorcycles. However, the total supply motorcycles consist of the TGM and the total electric scooters. The parameter values and the data sources of the related variables are listed in Table 26.7.

26.4.3 System Simulation and Result Interpretation

This paper simulates four scenarios as shown in Table 26.8. Scenario 1 is business as usual (BAU), which represents non implementation of any green transportation policy. Scenario 2 is to subsidize 6,000 electric scooters per year in 2011, 2012 and 2013. Scenario 3 is to annually subsidize 6,000 electric scooters (year 2011 through year 2013) and also stop licensing two-stroke gasoline motorcycles. Scenario 4 is a

Table 26.8 Four green transportation scenarios used in the case study

Scenario	Policy highlights
1	Business as usual (BAU)
2	Annually replace 6,000 existing gasoline scooters with electric scooters and implement the rebate scheme in year 2011, 2012, and 2013
3	Implement Scenario 2 and also terminate new license issuing of 2-stroke gasoline scooters
4	Implement Scenario 2 and also terminate new license issuing for all gasoline scooters

mandatory green motorcycle policy and also consists of subsidizing 6,000 electric scooters per year. In the current model, the running time set for the model is 20 years and starts from 2011. The time step for the simulation is 1 year. Scenario 4 is the best strategy for carbon reduction. Since the CF emission decreases to 11.9839 (Gg-CO_2) from 19.5778 (Gg-CO_2) and total oil consumption decreases to 4.1092 (1,000 kl). However, the total energy consumption will be significantly increased to 4,309.25 (1,000 kWh) and the cost of carbon reduction policies will increase.

Finally, this paper uses the results of above four scenarios to compare the carbon reduction goal of the Sustainable Energy Policy 2008, Master Plan of Energy Conservation and Carbon Mitigation 2009 and the Penghu low carbon island development project 2010. We test these four scenarios to see if the carbon reduction goals are achieved. The details of policies are shown in Table 26.9.

The result of four scenarios comparing with the carbon reduction goals is shown as Fig. 26.5. Scenario 2, which substitutes 6,000 electric scooters in 2011, 2012 and 2013 will not achieve all of the carbon reduction goals. When comparing scenario 2 with BAU scenario, the carbon reduction effect is insignificant because the demand of gasoline motorcycle is still large. Scenario 3 consists of substituting 6,000

Table 26.9 The goal of carbon reduction policies

Taiwan carbon reduction policies	Policy goal	Setting value
(A) Sustainable energy policy 2008	This policy plans that the national carbon emission will diminish to 2008s level during 2016–2020 and will go back to 2000s level in 2025	2008: 14.8291 Gg-CO_2
		2000: 13.0194 Gg-CO_2
(B) Master plan of energy conservation and carbon mitigation 2009	This policy plans that the national carbon emission will diminish to 2005s level during 2020 and will go back to 2000s level in 2025	2005: 14.4521 Gg-CO_2
		2000: 13.0194 Gg-CO_2
(C) Penghu low carbon island development project 2010	This policy plans that the regional carbon emission will go back to 50 % of 2005s level in 2015	2005: 7.2261 Gg-CO_2

Note revised from [6, 7]

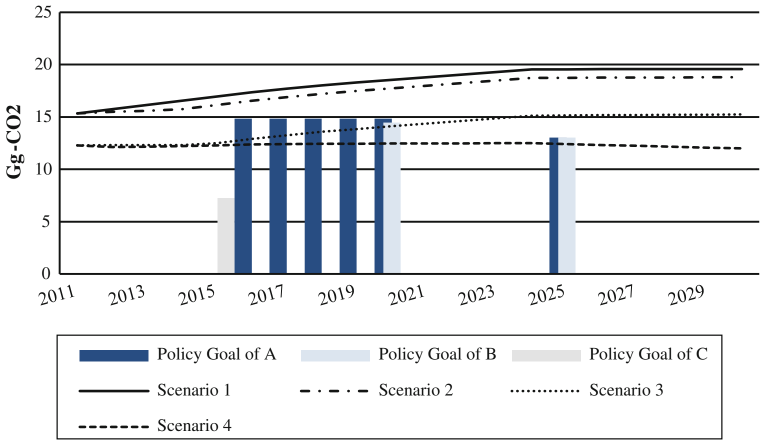


Fig. 26.5 The carbon footprint assessment of industrial sectors by EIO-LCA

electric scooters and limiting two-stroke gasoline motorcycles. The carbon emission of scenario 3 will diminish to 2008s level during 2016–2020 for Sustainable Energy Policy 2008 and go back to 2005s level during 2020 for Master Plan of Energy Conservation and Carbon Mitigation, but it cannot reach the 2000s level in 2025. After the government imports electric scooters for three years, the new four-stroke motorcycle license will gradually increase and increase the carbon emissions. Finally, scenario 4 consists of substituting 6,000 electric scooters, elimination of two-stroke gasoline motorcycles, and elimination of gasoline motorcycles. The carbon emissions for scenario 4 can achieve the planned carbon reduction goals.

26.5 Conclusions

Reducing emission of CO_2 is a much needed effort to slow the global warming problem. Thus, governments are trying hard to introduce effective green policies. If governments focus on the largest carbon emission sectors, the results can be significant. In this study, the national statistics shows that the transportation sector is one of the major sectors emitting CO_2 in Penghu. Taiwan's low carbon (Penghu) project, thus, introduces the green transportation pilot policy. SD models have been applied to analyze the effectiveness of investment costs and the corresponding benefits in four realistic scenarios. The result of scenario 4 appears to be the best strategy for carbon reduction. Therefore, the government must restrain the use of gasoline motorcycles while aggressively encourage people to buy electric scooters. As stated in [8], this research introduces a generalized methodology, combining EIO-LCS, LQ, and SD and following the paradigm of CE, which allows policy analysts to preview and assess the impacts of green policies prior their implementations.

References

1. Trappey AJC, Trappey CV, Hsiao CT, Ou JR, Li SJ, Chen KWP (2012) An evaluation model for low carbon island policy: the case of Taiwan's green transportation policy. Energy Policy 45:510–515
2. Trappey AJC, Trappey CV, Liu PHY, Hsiao CT, Ou JR, Chen KWP (2012) Location quotient EIO-LCA method for carbon emission analysis. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 367–377
3. IPCC (2001) IPCC third assessment report. http://www.ipcc.ch/ipccreports/tar/index.htm. Accessed 22 Nov 2010
4. Ministry of Economic Affairs (2011) Master plan of energy conservation and carbon mitigation. http://www.moea.gov.tw/Tapp/main/content/ContentImages.aspx?menu_id=3649. Accessed 1 Mar 2012
5. Trappey AJC, Trappey CV, Liu HY, Lin LC, Ou JR (2013) A hierarchical cost learning model for developing wind energy infrastructures. Int J Prod Econ 146(2):386–391
6. Trappey A, Trappey C, Lin G, Chang YS (2012) The analysis of renewable energy policies for the Taiwan penghu island administrative region. Renew Sustain Energy Rev 16(1):958–965
7. Bureau of Energy (2010) Penghu low carbon island development project, Bureau of Energy. http://www.moeaboe.gov.tw/Policy/PoMain.aspx?PageId=polistS. Accessed 29 Nov 2010
8. Trappey AJC, Trappey CV, Liu PHY (2013) Using a small island pilot plant approach to analyze low carbon emission policies. Carbon Manage 4(3):257–260
9. Carbon Trust (2010) Carbon footprinting. http://www.carbontrust.co.uk/cut-carbon-reduce-costs/calculate/carbon-footprinting/pages/carbon-footprinting.aspx. Accessed 29 Nov 2010
10. Minx JC, Wiedmann T, Wood R, Peters GP, Lenzen M, Owen A, Scott K, Barrett J, Hubacek K, Baiocchi G, Paul A, Dawkins E, Briggs J, Guan D, Suh S, Ackerman F (2009) Input–output analysis and carbon footprinting: an overview of applications. Econ Syst Res 21 (3):187–216
11. Chen F, Duic N, Alves LM, Carvalho MG (2007) Renewislands-renewable energy solutions for islands. Renew Sustain Energy Rev 11(8):1888–1902
12. Oikonomou EK, Kilias V, Goumas A, Rigopoulos A, Karakatsani E, Damasiotis M, Papastefanakis D, Marini N (2009) Renewable energy sources (RES) projects and their barriers on a regional scale: the case study of wind parks in the Dodecanese islands, Greece. Energy Policy 37(11):4874–4883
13. Demiroren A, Yilmaz U (2010) Analysis of change in electric energy cost with using renewable energy sources in Gökceada, Turkey: an island example. Renew Sustain Energy Rev 14(1):323–333
14. Uemura Y, Kai T, Natori R, Takahashi T, Hatate Y, Yoshida M (2003) Potential of renewable energy sources and its applications in Yakushima island. Renew Energy 29(4):581–591
15. Samso Energy Academy: Renewable Energy Denmark (2010) Updated 10 year anniversary report. http://www.onlinepdf.dk/Books/onlinepdf.aspx?onlinepdf=24054. Accessed 21 Nov 2010
16. Carnegie Mellon University (2010) Economic input-output life cycle assessment. http://www.eiolca.net/. Accessed 10 Nov 2010
17. Chang Y, Ries RJ, Wang Y (2010) The embodied energy and environmental emissions of construction projects in china: an economic input-output LCA model. Energy Policy 38 (11):6597–6603
18. Hendrickson CT, Lave LB, Matthews HS (2006) Environmental life cycle assessment of goods and services: an input–output approach. Resources for the Future, Washington
19. Chiang CA, Wainwright K (2005) Fundamental methods of mathematical economics. McGraw-Hill, New York, pp 115–123

20. Huang YA, Matthews HS (2008) Seeking opportunities to reduce life cycle impacts of consumer goods—an economy-wide assessment. In: Electronics and the environment (ISEE), San Francisco, 19–22 May 2008, pp 1–6
21. Alcántara V, Padilla E (2009) Input–output subsystems and pollution: an application to the service sector and CO_2 emissions in Spain. Ecol Econ 68(3):905–914
22. Machado D, Schaeffer R, Worrell E (2001) Energy and carbon embodied in the international trade of Brazil: an input-output approach. Ecol Econ 39(3):409–424
23. Peters GP (2008) From production-based to consumption-based national emission inventories. Ecol Econ 65(1):13–23
24. Ferguson TM, MacLean HL (2011) Trade-linked Canada-United States household environmental impact analysis of energy use and greenhouse gas emissions. Energy Policy 39(12):8011–8021
25. Acquaye AA, Duffy AP (2010) Input–output analysis of Irish construction sector greenhouse gas emissions. Build Environ 45(3):784–791
26. Trappey AJC, Trappey CV, Hsiao CT, Ou JR, Chang CT (2012) System dynamics modeling of product carbon footprint life cycles for collaborative green supply chains. Int J Comput Integr Manuf 25(10):934–945
27. Ju L, Chen B (2010) An input-output model to analyze sector linkages and CO_2 emissions. Procedia Environ Sci 2:1841–1845
28. Forrester JW (1968) Principles of system. MIT Press, Cambridge
29. Wang J, Lu H, Peng H (2008) System dynamics model of urban transportation system and its application. J Transp Syst Eng Inf Technol 8(3):83–89
30. Jin W, Xu L, Yang Z (2009) Modeling a policy making framework for urban sustainability: incorporating system dynamics into the ecological footprint. Ecol Econ 68(12):2938–2949
31. Han J, Hayashi Y (2008) A system dynamics model of CO_2 mitigation in China's inter-city passenger transport. Transp Res Part D: Transp Environ 13(5):298–305
32. Miller R, Blair P (1985) Input-Output Analysis: Foundations and Extentions, Prentice-Hall, Inc, New Jersey
33. Penghu Country Government (2011) Geographic location. http://www.penghu.gov.tw/eng/00home/home.asp. Accessed 5 Mar 2011
34. National Statistics, Republic of China (2010) Input coefficients table at producers' prices (Table 2). http://eng.stat.gov.tw/ct.asp?xItem=25743&ctNode=1650. Accessed 22 Nov 2010
35. National Statistics, Republic of China (2007) Manpower survey results in 2006. http://www.dgbas.gov.tw/ct.asp?xItem=18182&ctNode=3247
36. National Statistics, Republic of China (2010) Agriculture, forestry, fishery and husbandry census of 2005 statistical analysis, http://eng.stat.gov.tw/lp.asp?ctNode=1633&CtUnit=783&BaseDSD=7. Accessed 10 Nov 2010
37. National Statistics, Republic of China (2008) Industry, commerce and service census of 2006 statistical tables. http://eng.stat.gov.tw/lp.asp?ctNode=1624&CtUnit=774&BaseDSD=7
38. Lin SM, Huang TH (2006) Resource utilization model update and maintenance with the analysis and policy simulation. Authorized project by council for economic planning and development

Chapter 27
Sustainable Mobility

Alain Biahmou

Abstract Considering sustainable mobility, the electrical powertrain of road vehicles has an increasingly significant role. Besides delivering benefits in air and noise pollution, it encompasses huge challenges in practical usability, reliability and total costs of ownership combined with novel models of exploitation. Therefore, sustainable mobility is a typical field of application for Concurrent Engineering. The design of electric vehicles requires bringing components from different domains together in order to integrate them in the overall vehicle concept. The domains involved utilize their own specific methods, processes as well as software tools in order to create partial models of an overall system. This leads to dependencies between several disciplines and, therefore, to the need to track the impact of model interactions to avoid data inconsistency as well as design errors. The focus of this paper lies on the project "Process Chain Battery Module" that has been conducted at EDAG Engineering AG to capture the challenges related to the electrical battery when designing electric vehicles. Thermal management, which is one of the critical challenges to be tackled in the area of electro mobility, is discussed and solution approaches are presented. Requirements are defined and linked with functional analysis as well as geometrical, behavioral and FEM models. Thus, changes can be traced from each partial model back to the initial requirements. Interface management between the domains and partial models is realized to enable an analysis of the entire vehicle. Complex simulations are performed in a very early stage of development in order to determine the range of an e-vehicle model (EDAG Light Car).

Keywords Sustainability · Systems engineering · Mechatronics · Battery simulation · Thermal management

A. Biahmou (✉)
EDAG Engineering AG, Reesbergstr. 1, 36039 Fulda, Germany
e-mail: alain.biahmou@edag.de

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_27

27.1 Introduction

Most transport vehicles are powered by an internal combustion engine which is the most widespread power supply for cars nowadays. Although the development of combustion engines may bear further improvement potential, it is important to remark that using combustion engines is related to side effects such as the emission of carbon dioxide and noise emissions that contribute to environmental pollution. Furthermore, the increasing scarcity of fossil fuels and the more and more stringent laws for environmental protection lead to the necessity of developing alternative solutions for powering cars.

The analyses that have been performed during the project described in this chapter are focused on car concepts, especially the powering of a car with a battery, since it is the most important component of electric cars. The battery significantly influences the maximum reachable distance, weight, power, price and life duration of an electric car [1].

The powertrain of an electric vehicle differentiates itself from the powertrain of conventional cars. Electric motors can start from stop position and operate efficiently over a wide speed range. Therefore, it is not necessary to integrate a clutch in the vehicle design, as it is the case in internal combustion engine vehicles. For hybrid vehicles, the internal combustion engine is to be coupled with the electric motor; this makes the transmission more complex [2].

The electric car is the most popular alternative solution to cars powered by combustion engines. The battery technology and the electrified powertrain are the major innovations of electric cars compared to conventional cars powered by a combustion engine. Existing concepts of electrical powertrains are based either on a battery as single power supply, hybrid electric power supply as well as micro fuel cells. Cars powered by a battery are characterized by a limited overall range and a relatively long battery recharging time. Besides, additional systems inside the car such as cooling devices, but also external factors such as low temperatures and the shape of the terrain may significantly reduce the maximum reachable distance [1].

In fact, the driving range and the battery power are the most important parameters of an electric car from a customer perspective, even though considerations regarding the infrastructure (e.g., power supply stations) also play an important role. The vehicle price and the driving experience are certainly predominant factors; however, the vehicle price is directly impacted by the price of battery packs. In order to significantly increasing the driving range, the number of battery packs and, therefore, the total car costs are also to be increased. To decouple the relationship between driving range and vehicle costs, a study proposes inductively coupled power transfer (ICPT) as a potential solution [3]. ICPT is a technique that can transfer power without physical connection between vehicle and energy source. The objective of this approach consists in increasing the range of an electric vehicle without substantial impact on the weight or cost of the vehicle. The power track is to be located into a road surface, while the vehicle receives the induced energy through an integrated Pick-up [3]. The vehicle is to be directly powered with electromagnetic inductive energy, which alternatively can be used to charge the vehicle battery [4].

Battery cost and the limited range of electric vehicles also can be improved using a range extender, which consists of an on-board fuel converter that converts a liquid fuel, such as gasoline, into electrical energy whilst the vehicle is driving. This approach has been demonstrated with a compact class vehicle [5]. Although the objective of carbon-free emission cannot be reached while the range extender is contributing to the drive, emissions can be reduced, in particular for short runs, improving the usability dramatically.

The difference between a conventional and an electrical car is not limited to their powertrains, since the development of electric cars opens various opportunities for new concepts, like using singular electro motors for driving each wheel or very complex energy business models which describe new interaction models of power providers with owners of electric cars. Such a business model may consist of owners of electrical cars buying or selling electricity back to an energy provider not only for stabilizing the current network but also to make profit, depending on the day time and the need. Besides, the intensive use of electric cars may lead to a modification of the architecture of current networks as well as the adaption of cities, which would have to integrate a suitable infrastructure, for instance load stations.

It is important to get a whole car perspective in order to gather the mutual influence of relevant parameters [6]. Sharing of status information as well as technological data has to be observed for realizing an efficient parallelization of processes, which is a core principle of concurrent engineering.

In fact, the questions that have drawn the attention concern determining an efficient way of working when it comes to develop electrical vehicles. Since electric cars concepts are different from conventional ones, tools and methods as well as interfaces between the different disciplines involved might also be different or at least imply additional facts that are to be taken into account to keep the cutting-edge advantage of EDAG Engineering AG as one of leading engineering companies in Germany [7].

This chapter continues with a brief introduction to sustainability (Sect. 27.2), whereby its perception in the automobile industry is emphasized. Selected research works that contribute to sustainability are highlighted (Sect. 27.3). A case study is addressed (Sect. 27.4), in which Systems Engineering has been applied in order to realize the technical design of an electric car battery, the most important additional module of an hybrid or electric car, contributing therefore to sustainability. The case study is realized following the RFLP methodology. That approach describes developing complex products by starting with a requirements model, then deriving a functional model, from which a logical and a physical model can be created. Discussion and reflection to concurrent engineering is made and some perspectives for the case study are mentioned (Sect. 27.5), followed by conclusions and outlook in Sect. 27.6.

27.2 Sustainability in Car Development

Although the term sustainability sounds like an invention of the two past decades, it has been first used by Hans Carl von Carlowitz in his book titled “Sylvicultura Oeconomica. Die Naturmäßige Anweisung zur Wilden Baum-Zucht”, published 1713 in Germany. The term sustainability has gained a more particular attention since the 1992 Rio Earth Summit with the motto sustainable development. The core idea has consisted of preserving natural resources for maintaining the quality of life of future generations [8].

Sustainable vehicle design process, therefore, should include criteria such as technical performance, design, vehicle production, cost, quality and so on. In order to achieve greater benefits in terms of minimized environmental load and cost, sustainable design principles are to be integrated into the development process [9].

In the automotive industry, sustainability has evolved from its initial understanding as ecological development and production of vehicles to a holistic concept, which integrates the reduction of pollution and resource consumption, the quality of life of the population as well as economic success of enterprises. Nowadays, the original equipment manufacturer (OEMs) are not only interested in fulfilling legal requirements over environment protection, since sustainability has become a purchasing argument. Furthermore, corporate sustainability is an important factor for business success, because investors can make their decision according to sustainability rankings which are provide for instance by the Dow Jones sustainability index (DJSI).

Enterprises have identified many factors which contribute to sustainability, such as manufacturing processes, the optimization of combustion engines, alternative engines, materials and vehicle architectures [10].

27.2.1 Manufacturing Processes

Some works have presented technologies that enable the manufacturing of materials with specific characteristics, such as a light weight, leading to reduced resource consumption. Other research approaches focus on manufacturing material with important chemical properties.

Bruckmeier and Wellnitz present a pultrusion technology that enables the manufacturing of profiles with specific short-term mechanical properties such as stiffness, strength, elongation to failure, shear strength and impact resistance. Pultrusion is a continuous method of manufacturing various reinforced plastic shapes of complex cross sections. The elaborated fiber glass-reinforced pultruded polyurethane may help reducing wall thickness and, therefore, component weight, even though the knowledge of its long-term mechanical behavior is limited [11].

Cannon et al. [12] have presented a technology for creating a microstructure on the surface of parts in order to provide them with a superhydrophobicity ability.

Superhydrophobicity is the ability of some surfaces to imitate the water repellency of lotus leaf. Superhydrophobic surfaces can also exhibit “self-cleaning” properties, which are useful for automobiles because droplets that roll off of the surface carry away particles that are larger than microstructure spacing. The materials that can be microstructured by the approach presented include stainless steels, tool steel, nickel, titanium, copper and carbide steels.

Volatile organic compounds (VOC) are organic chemicals that are able to evaporate and, therefore, enter and pollute the air. They are emitted by cars even at switch-off state and enter the surrounding air. Due to the health risks associated with VOCs, limit values for VOC emissions have been formulated in guidelines such as 2004/42/EG/ by different institutions worldwide (e.g., EU, AgBB, AFS-SET, California Department of Public Health). The objective of some research work consists of reducing VOC emissions. A research approach following that principle is a waterborne pretreatment technology for Direct Glazing, to be applied in Automotive [13].

27.2.2 Enhancing Sustainability with Optimized and Alternative Energy Sources

Many contributions to sustainability are dealing with the optimization of fuels to reduce environment pollution, while others are proposing alternative energy options to replace fossil fuels with renewable energy sources. Well-known examples are Bio-fuels and electricity produced by solar, wind and geothermal energy, which are largely used nowadays.

Batteries are the energy source that have been mentioned most in the context of electric vehicles. The requirements of EV and hybrid electrical vehicles are high specific power, high specific energy, and high charge acceptance rate for recharging and regenerative braking, long calendar and cycle life [2].

Although many research works have focused on improving battery properties, battery development has not yet reached the stage which would be necessary to power a car on very long ranges. A serious alternative for battery is the fuel cell.

The use of methanol in methanol-to-gasoline, -diesel and -kerosene processes to synthesize drop-in fuels that can fully decarbonize the known forms of transport have been discussed. These discussions have been made on the assumption that the vehicles must not be changed significantly. Furthermore, methanol has been presented as an interesting transport fuel for spark-ignition (SI) engines, because it is synthesized from sustainable sources. The configuration of ternary blends out of the three components methanol, ethanol and gasoline as well as using methanol consuming fuel cells for range extended electric vehicles have been analyzed [14].

The optimization of existing fuels goes beyond the research of substances that might be blended. The compositions of blends are subject of further studies. Bunting et al. have evaluated diesel range fuels using a homogeneous charge compression ignition (HCCI) engine. The analysis has included bio-diesel blends with differences like oxygen content, iodine number, cetane, boiling point distribution, chemical composition, and contained nitrogen. Fuel and engine control variables are used as input variables of the experiment, while emissions (NOx-Nitrogen oxide, smoke), fuel economy (ISFC) and control information (intake temperature) are output variables. Fuel with lower nitrogen and oxygen, lower cetane and lower aromatics offered the best results [15].

The most promising option for realizing a zero-emission objective is to use a fuel cell, which is an electrochemical engine. Unlike a combustion engine, the fuel does not burn; it reacts with air. In order to carry out this process, hydrogen, methanol, ethanol, natural gas, as well as liquefied petroleum gas, are often used as fuels. The most relevant fuel cell technologies are molten carbonate (also known as direct fuel cell—DFC), proton exchange membrane (PEM) and solid oxide. Additional to electricity, DFC produces heat as by-product, PEM and solid oxide technology generate water and heat as waste products [16].

An alternative fuel that offers carbon-free transport is, therefore, hydrogen since its reaction with oxygen provides water. Some OEMs are interested in using hydrogen fuel cell (H2FC) that produce electrical energy to drive an electromotor that in turn propels the vehicle. An alternative approach consists of using an internal combustion engine (H2ICE), which uses hydrogen as fuel. Using H2FC helps obtaining a higher efficiency, since the efficiency of the H2ICE approach is limited by the heat phases that are included in the engine cycles. On the other hand, cost and weight of H2ICEs are more advantageous and they can run on fuel of less good quality (e.g., impure fuels). A comparison of both systems emphasizes the negative impact of the mass of fuel cells on the whole vehicle mass, but also the fact that not only isolated factors such as thermodynamic efficiency and power density are to be taken into account when comparing vehicles. All relevant system parameters are to be considered [17].

There are already commercial hydrogen fueling stations using the hydrogen that has been produced either with geothermal power by stripping hydrogen out of water molecules or with wind power [16]. Although hydrogen—to power the fuel cells—is appropriate as alternative energy option, one of its short-comings is its small volumetric energy density, which complicates its storage significantly and, therefore, its transport, distribution and usage in series-production vehicles. This has led manufacturers to using alternatives such as solar-powered hydrogen stations or the application of technologies on-board to extract hydrogen out of other substances and provide it to fuel cells. Thus disadvantages of providing hydrogen to vehicles can be solved using methanol [14, 16].

However, the reformer (hardware) used to reach this objective can be very heavy. It can also produce a certain amount of carbon dioxide as by-product and, therefore, would impact the energy balance. Additionally, ammonia can be used as alternative fuel for internal combustion engines, since ammonia is made out of a large fraction of hydrogen. Ammonia can be liquefied easily and a suitable synthesis technology is well established. Tests have been conducted using an autothermal cracker (ATC) to dissociate hydrogen and nitrogen from ammonia and provide them to a single-cylinder engine. The results have shown that a stable combustion can be achieved [18].

An objective of many manufacturers is making the fuel cells smaller, lighter and inexpensive. The advantage of a fuel cell over a battery is that the potency of batteries may decrease. In contrary, fuel cells keep producing power as long as the fuel and oxidant supplies are maintained. Another research approach is the use of microbial fuel cells (MFC). This technology consists of generating electricity out of organic material available in waste-water. Therefore, biomass can be used to generate hydrogen and ethanol that can be used to produce electricity for powering an engine. Advantages of using MFCs go beyond the reduction of emissions, because the cleaned water may be used for other purposes. The MFC technology for power generation is certainly not yet mature for use in serial vehicles, but it could become a serious option in the future [16].

A rather new method for generating hydrogen for fuel cells consists of using algae, from which sulfur is separated and hydrogen is generated. This method is still explored in prototypes and is not yet appropriate for an industrial deployment.

An economic and environmental comparison has been performed on conventional, hybrid, electric, H2FC, hydrogen-fueled ICE and ammonia-to-hydrogen vehicles. The comparison has shown that hybrid and electric vehicles are more advantageous than the others. Electric vehicles may have a better balance than hybrids depending on the conditions of electricity generation, for instance when it is produced from renewable energy sources [19].

27.2.3 Enhancing Sustainability with Materials

Some contributions to sustainability consist of optimizing properties of existing materials [20], but also of substituting them with alternative ones (e.g., biologically degradable materials). Besides, optimization can be achieved through the modification of the structure of a technical system. Many studies following that approach are based on lightweight engineering, which may help achieving a very high material and energy efficiency. Composite materials generally are used for that purpose in the mobility industry.

Composite structures can be used in the aerospace industry to reduce weight and, therefore, fuel consumption while simultaneously improving the structure of an aircraft. Basic structural elements such as plates and shells are often used in airplane structures to absorb vibration [21].

A further study explores a combination of lightweight design of a seat shell developed with respect to the holistic design approach of the so-called “Dresdner Modell” with a manufacturing process for high-volume production of textile-reinforced thermoplastic materials. A prospective resource-oriented product assessment of the seat shell is performed and its lifecycle (in fact energy and material inputs of the different lifecycle phases) is compared with that of a corresponding steel seat based on resource demand. The results confirm that the lightweight seat shell can significantly reduce resource consumption [22].

Further studies have focused on applying lightweight design to obtain a mass reduction, whereby several design parameters are varied and simulated in order to search and manufacture a suitable, totally new material. Examples have consisted of running finite element analysis (FEA) and optimizations to search suitable materials, configuration and topology of a side door [23].

A similar study follows the objective of creating a new composite material [24].

27.2.4 Enhancing Sustainability with Optimized and Alternative Drive Systems

Due to the obstacles in producing zero-emission vehicles on a large scale, the reaction of OEMs has been to elaborate intermediate solutions, which are hybrid vehicles. These generally combine a gasoline engine with electric motors. Hybrid vehicles may be classified in serial, parallel, power split and combined hybrids. Serial hybrids use a combustion engine that is coupled to a generator to provide energy to the driving electromotor and simultaneously to the battery. Parallel hybrids combine a combustion engine with one or more electrical motors, whereby both motors can interact, for instance to provide the necessary power when driving up- and downhill or for recharging the battery. This hybrid type has the highest potential to reduce fuel consumption. Power split hybrids enable a separation of the mechanical power to be transmitted into a mechanical and an electrical path.

A particular case is the combined hybrid, which enables both serial and parallel operating. Depending on their degree of hybridization, there are micro, mild, full and plug-in hybrids. The latter additionally enable external battery recharging requiring a higher battery capacity and, therefore, an increase in costs. Nevertheless plug-in hybrids might be a transitional solution to the electric vehicle [25]. While the first hybrid vehicles were equipped with a battery for powering the electromotor, latter hybrid cars combine a battery and fuel cells [16].

Shortcomings of hybrid vehicle are due to the fact that the system is more complex since two motors are integrated. This requires more skills for maintenance than for conventional cars. However, hybrid vehicles can substantially reduce emissions in urban traffic.

Rotatory piston machines (RPM), which transform stored chemical or physical energy into rotary motion or vice versa, are presented in a further study. RPM technology may offer a greater power density and a corresponding increase in efficiency, compared to customary aggregates. Additional benefits are a simultaneous reduction in built space volume and weight of the engine. An RPM turns slower than a conventional engine of the same power, for example, a standard reciprocating piston machine or a Wankel engine. This impacts the combustion process and, therefore, the quality and quantity of the exhaust gases and pollutant emissions [26].

27.2.5 Summary Evaluation

The technologies presented above are very interesting and even though some research work seems fundamental and mono-disciplinary, it is noticeable that the results presented imply concurrent engineering thinking. If for some works, other disciplines come in with evaluation, comparison and testing, for other one, the are part of the initial concepts, especially when innovative concepts are to be investigated. In these cases, the infrastructure for testing also must be developed parallel to the core innovation, for instance the material or the fuel that should help enhancing sustainability.

The works that have been performed to enhance sustainability with specific manufacturing processes show a cooperation at less between the disciplines manufacturing engineering, material engineering and nano- and high-precision technologies in chemistry. The same, works that investigate alternative energy sources often are based on a multidisciplinary approach, for instance when it comes to determine the impact of fuel cell mass on the whole vehicle, or when it comes to develop models for transporting and distributing hydrogen. Furthermore, it is assumed that concurrent engineering plays an important role in projects groups that have to develop concepts involving biological knowledge, chemistry and mechanical engineering. This is the case when for instance, fuel is to be gained out of algae or any organic material.

In the practice, car makers already are exploring the federation of many factors that enhance sustainability. For instance, combining conventional light-weight design with the substitution of steel parts by composites structures as well as using an hybrid propulsion. However, this way of thinking should be emphasized and the

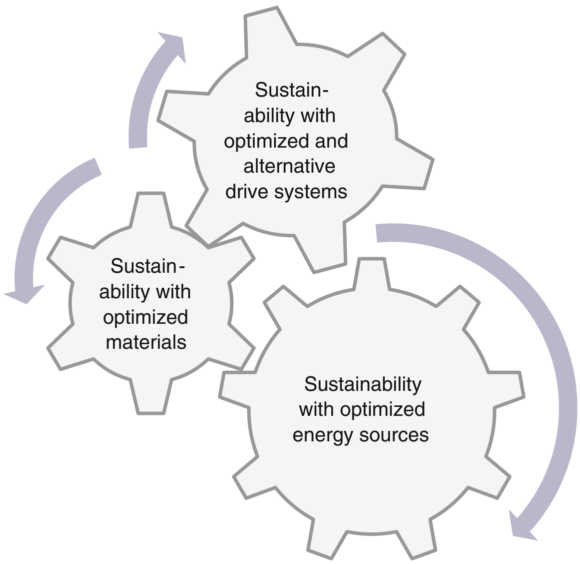


Fig. 27.1 Different ways to realize sustainability, which are considered in this work

impact of the different approaches for enhancing sustainability should be assessed in order to ensure synergy and strategic orientation of the multitude of research works that are currently performed in academic and industrial institutions. It is a matter of evidence that the different approaches can impact each other as shown in Fig. 27.1.

27.3 The Generic Car Development Processes

Even though the principles of driving as well as the architectures of cars have not significantly changed in the past decades, car development processes have been subject to significant modifications [27]. The components to be designed have become very complex and have integrated functions which imply knowledge in different disciplines. Therefore, many disciplines develop components, a process which in the past used to be aimed at charging a single engineer.

In order to determine the impact that electric cars may have on existing processes, it is important to draw a picture of product development processes as well as tools that are involved. First of all, product strategy defines a profile of a new car according to the relevant brand profile, which itself is intended to impact the long-term perception of customers. A specific car profile provides details of factors such as segment, the targeted population as well as characteristics such as level of comfort, functionality, size and so on.

The next development phase is the feasibility study that includes economical as well as technical feasibility based on a target framework. The latter integrates essential factors such as weight, variants, technical function, equipment, innovation, production and assembly, quality, services and administration. These studies consist of determining whether the envisioned car project is to be conducted or not [28].

Three-dimensional virtual and real prototypes are involved in the concept phase in order to check whether the car to be developed will meet the requirements formulated during previous development phases. First a clay model is created, and then reverse engineering techniques are applied to digitize its shape. Some research works have been presented in the past to optimize this phase while replacing real clay models with virtual prototypes (see Chap. 13), contributing, therefore, to sustainability. The objective has been the representation of virtual material in an immersive environment, which integrates realistic interaction for sculpting the material with virtual modeling tools similar to real ones used by designers. The interaction was realized by a specific device, which has been connected to two force feedback devices in order to let the user feel the forces arising from the virtual process [29, 30].

The conventional process has still priority in the industry. Therefore, virtual surfaces (CAD surfaces) which are used for conceptual studies are still derived from the clay model and processed by designers to obtain “class A” surfaces. Further analyses regarding ergonomics, aerodynamics, handling and production are based on these virtual surfaces to validate the car concept. The concept phase provides a product, production, sourcing, sales and marketing as well as a services concept as deliverables.

The series development phase is triggered by the end of the concept phase. Detailed design is performed and releases (for instance tooling release) are fixed as milestones. Tools, equipment as well as processes for product manufacturing are developed in this phase. Therefore, there are interactions between the product and tooling development. Components and systems are integrated into the whole car to verify that the requirements are fulfilled. For this purpose, real prototypes (pre-series and series vehicles) are built. Doing so provides a first impression of the suitability of the planned processes and prototype tools. The start of production (SOP) follows a positive evaluation of first pre-series vehicles. After that milestone, the series support strives for performing short-term optimization (e.g., quality) as well as fulfilling short-term requirements [28].

Figure 27.2 shows a representation of a simplified car modeling process, in which the stages of product development, which are to be investigated in the frame of the battery module project, are highlighted.

The battery module is an example of sustainable engineering for mobility, that will be detailed in the next chapter as case study. Similar case studies have been addressed in the past [31, 32].

Dhameja has presented an approach for simulating and validating the development of batteries for hybrid and electric vehicle applications. A performance analysis integrated with a computer-based simulation provides a baseline for the battery pack in real-world conditions. Among others, data related to power draw, engine torque, speed and acceleration of the vehicle are analyzed [31].

Peng et al. apply the concept of an open architecture product (OAP) for the development of a miniature electric car. The modification or adaptation of product modules for different requirements during the product lifecycle is the main goal of the study. The vehicle functions are mapped to design parameters using a functional structure. A common platform, functional module as well as personal feature elements are integrated through mechanical, electrical and software interfaces to realize an OAP [32]. Therefore, customers may customize their vehicle for enriching the original functions. Especially for designing electric cars, adaptability and sustainability are major factors.

However, the case study realized with EDAG Light Car is a much more comprehensive case, in which all relevant disciplines for vehicle development have been involved, applying therefore concurrent engineering on an industrial level.

Requirements for an electric car have been formulated and a development process was simulated in order to identify necessary interfaces as well as processes, tools and methods. The geometrical model of EDAG Light Car, which is a concept model of an electric vehicle, has been involved to build a proof-of-concept.

The approach that was followed throughout the elaboration of the process chain battery module is Systems Engineering. Therefore, understanding Systems Engineering basics is necessary to reconstruct the single activities which have been realized. However, a detailed excursion in that direction would exceed the scope of this chapter. A survey over Systems Engineering is presented in Chap. 9 of this book.

27.4 Process Chain Battery Module

The main objective of the project "process battery module" is contributing to sustainability through the use of an alternative, green power supply for vehicles. The first step consists of the preparation of the EDAG Light Car Model to pilot the stages of the car development process, which have been presented in Fig. 27.2. Necessary interfaces and adapters for information exchange between the disciplines involved are to be identified and processes as well as methods which enable an integrated process from the requirements definition to the stage "CAD and Simulation Body" (see Fig. 27.2) are to be defined.

In most cases, the different disciplines involved in the product development use tools that are not connected to each other. This situation is due to the fact that the different tools are proprietary and sometimes there is no standard specification that can be implemented as software interfaces or adapters to support the transmission of information from one System to another. The consequence of this fact often is the isolation of product development stages (see Fig. 27.3), which normally should be connected for tracking purposes and, therefore, managing complexity. Thus, the list of requirements commonly is not linked to functions, although functions are derived from requirements.

From an information systems (IS) point of view, this challenge may be tackled by changing the traditional application landscape of companies from a set of autonomous systems to a service-oriented architecture (SOA) [34], which enables a

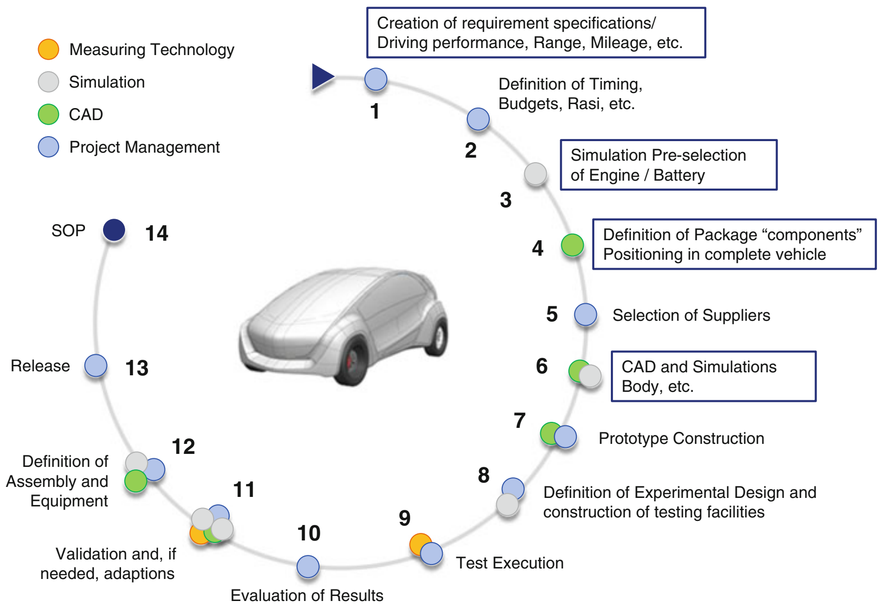


Fig. 27.2 Simplification of the car development process

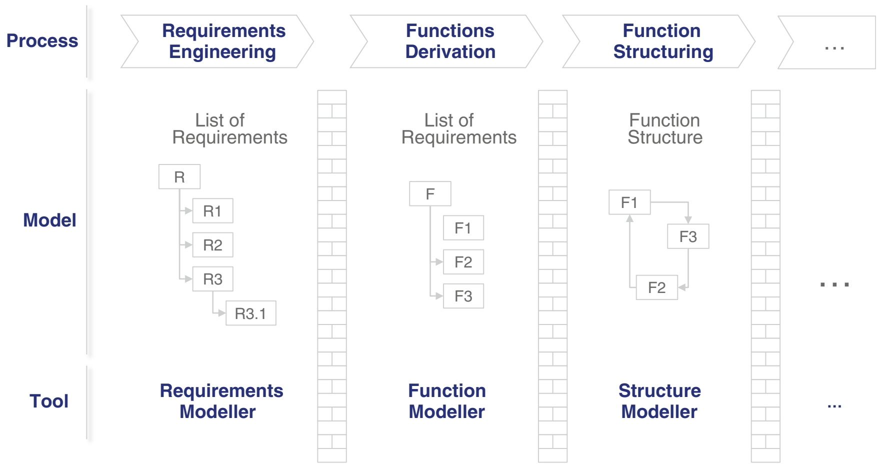


Fig. 27.3 Traditional systems engineering approach according to VDI2206 [33]

loose coupling of applications used for product development (see Chap. 4). Such a SOA can help realizing a continuous data sharing between the disciplines involved, rather than translating and importing information for each system. For this purpose, system vendors may be of importance by providing necessary interface specifications or even Application Programmable Interfaces (APIs) to enable the implementation of adapters through third parties. This objective has been emphasized by the Codex of product lifecycle management (PLM) Openness, which is an initiative of the PROSTEP iViP Association defining criteria for the openness of IT solutions used in PLM. The main target of CPO consists in bringing IT vendors to make possible the fact that data created within a company can be used throughout the entire product lifecycle (see Sect. 21.6.1) [35].

For this purpose, interoperability for an efficient collaboration, the ability to be integrated in existing IS infrastructure, functional extensibility, documented interfaces, standards as well as comprehensibility of the architecture have been adopted among others as criteria for the openness of an IT solution. The CPO-community is made up of tree types of enterprises: IT solution providers, IT integrators and customers. IT Customers can take benefit from requirements or expectations formulated withing the CPO-community to negotiate contracts with IT solution providers.

27.4.1 Methodology

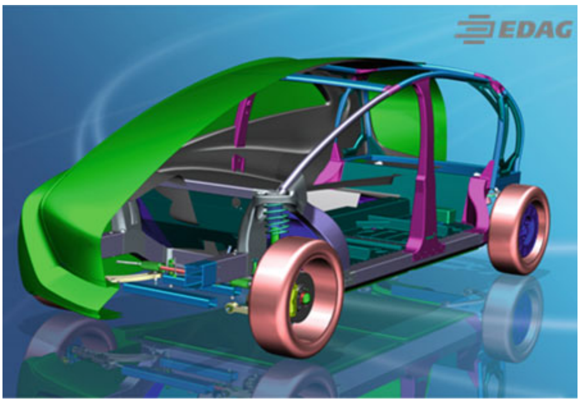
Many design methodologies have been defined in diverse VDI Guidelines (Association of German engineers), of which the VDI Guidelines 2221 and 2222 are some of the most known. VDI is the Association of German Engineers, that supports, promotes and represents them in their work. The members of VDI work on a voluntary basis in many project groups, that elaborate Guidelines for the engineering community [36]. VDI 2221 describes a general approach to develop technical systems that may be a machine, a plant, or software. The main development phases mentioned are task clarification, conceptual design, embodiment design and detail design. VDI 2222 gives a detailed explanation of methods that can be used to implement the different design phases described by VDI 2221 [37]. Pahl and Beitz describe the design process similar to VDI 2221 and provide deep details on the single steps to be followed, for instance the elaboration of a functional structure and the procedure for modeling its functions [37, 38]. The VDI 2206 provides a more suitable methodology for developing mechatronic products based on the V model [39].

For this study, a simplified methodology was followed, whereby requirements are defined and the relevant functional structure is derived accordingly [40]. The defined functions are used as input for the logical structure that represents a technological implementation of the functional structure. The next view on the product to be developed consists of the geometry representation, that is, the physical product form. The latter can be used, for instance, for performing a FEA to predict the impact of the battery (e.g., thermal influence) on surrounding parts.

27.4.1.1 Analysis Model

Input for the analysis of the process chain of the battery module has been the EDAG Light Car (Fig. 27.4), which is a vehicle concept for mobility of the near future. Some of the particularities of the EDAG Light Car are the light weight for energy efficiency, the light which is used as a central element to display functions and communication as well as the architecture in compliance with electric vehicle requirements. Furthermore, scalability has been an important requirement; therefore, the platform of the EDAG Light Car can be varied to build a car family. The integrated electric drive enables a drive-line variation for urban traffic. The battery cells are grouped into modules and positioned on the sandwich floor.

Fig. 27.4 EDAG light car



Since the development of electric vehicles differs from that of conventional cars, due for instance to the different architectures of both vehicle types, a different and appropriate way of working is required. From a tools and methods point of view, it is important to define processes and identify the tools that are suitable to tackle that challenge. Especially the influence of the battery on car behavior and structure is to be analyzed and taken into account, since it is one of the key factors of a battery-driven electrical vehicle. This calls for

- the traceability of requirements to subsystems,
- a central and whole vehicle analysis to identify the global impact of requirement parameters,
- an approach for simulation data and thermo management, which both are critical for electric vehicles,
- an intensive, distributed and multidisciplinary collaboration,
- an impact analysis of the different partial models and the management of interfaces between the different disciplines as well as subsystems in order to oversee the whole system; impacts of changes and parameters on subsystems are to be tracked and considered, and the
- transparent providence of information and technological data for process parallelization, which means that the disciplines should indicate the maturity of data in order to help other disciplines assessing the quality of data they are sharing.

The battery module is a chemical, thermal, electrical and mechanical system and is, therefore, to be developed with principles of systems engineering. The approach followed consists of defining a realistic process cycle from the definition of requirements to manufacturing. The tools and processes used for designing a conventional car were to be analyzed regarding the specificity of electrical vehicles.

For the study described in this chapter, battery cells were associated to a simplified, designed heat sink, building the so-called battery module. The modules were positioned on the floor of EDAG Light Car, allowing a rational use of space. The cell type used for this study is not typical for electric vehicles, but it often is used for applications in small serial or experimental vehicles.

One of the most important factors for performing a requirements analysis has been the ability to keep the traceability of requirements in line with the different partial models of the battery module. Doing so helps engineer efficiently assess whether some changes remain within acceptable ranges or not. Therefore, rework and loops between the disciplines involved can be reduced.

Selected requirements have been formulated for the battery module to be analyzed. Among others, the following requirements were of importance:

- Car range in km,
- Maximal velocity in km/h,
- Load time of the battery,
- Lifetime of the battery, and the
- Acceleration of the electric vehicle.

These requirements have been integrated manually into a product data management system, but it would have been possible to import them from a document or from an external requirement management tool, whereby it would have been necessary to implement an appropriate interface.

27.4.1.2 Function Structure

The main objective of building a function structure is to manage the complexity using the divide and conquer principle. Starting from the requirements, the main function as well as sub-functions can be recognized. The art consists of connecting the different identified sub-functions through logical relationships (e.g., input, output, sequence) in order to obtain the main function. Therefore, the engineer will have to look for solutions for implementing the sub-functions, which are less complex than the main function.

The function structure is solution-neutral, meaning that it defines a very high-level concept of how the main function may be realized, but it does not provide a solution like a physical component.

27.4.1.3 Behavioural Simulation

There are two main approaches for elaborating a logical model, which are signal-based modeling and object-based modeling. Signal-based modeling is characterized by explicit relationships that are defined to determine the output signal from specific input. Light modification of the physical model may lead to a major adaption of the system model. Object-oriented modeling enables inclusion of system components that are described implicitly. Virtual components correspond to real ones. For instance, there is a battery model corresponding to the real battery. Local modifications can be performed within components (e.g., battery model), which also can be replaced without modification or adaptation of the whole model. This approach is followed by the module CATIA Systems, which has been used for this study [41].

Thus, the logical structure created is represented by its logical, object-oriented components, among others the battery model and the mechanic model. These models are connected with each other using signals and dependencies (Fig. 27.5). The mechanic model is the geometrical representation of the EDAG Light Car, whereas the battery model is made out of a set of Modelica libraries developed by an external partner. Furthermore, the logical system structure is connected to requirements.

The driving cycle for the analysis is defined in Block 1. A driving cycle is a set of conditions under which a vehicle is meant to be driving to assess properties such as energy consumption or emissions. In the practice, it is a sequence of points that represent the speed of a vehicle versus the time. It is important since it helps determining the performance of different vehicles with comparable criteria. The new

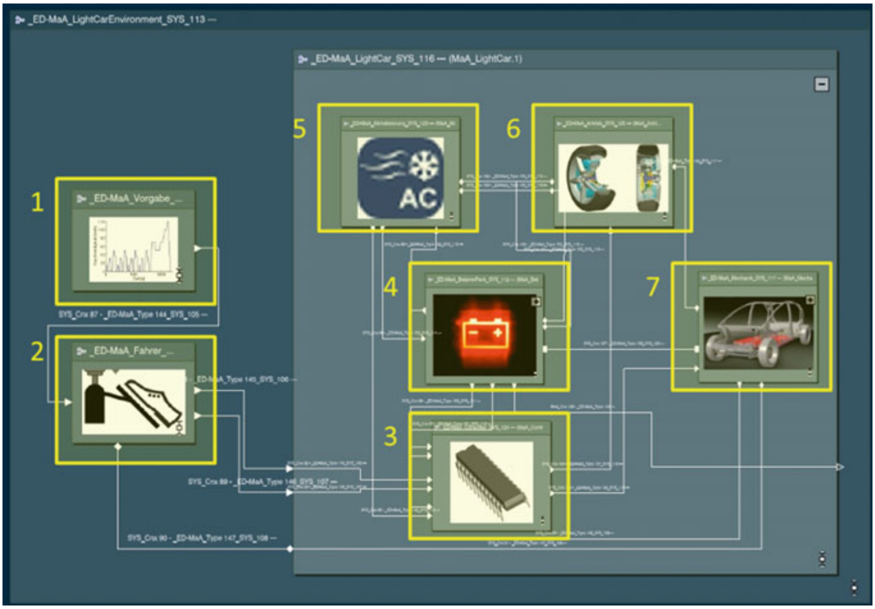


Fig. 27.5 Logical model of the whole system

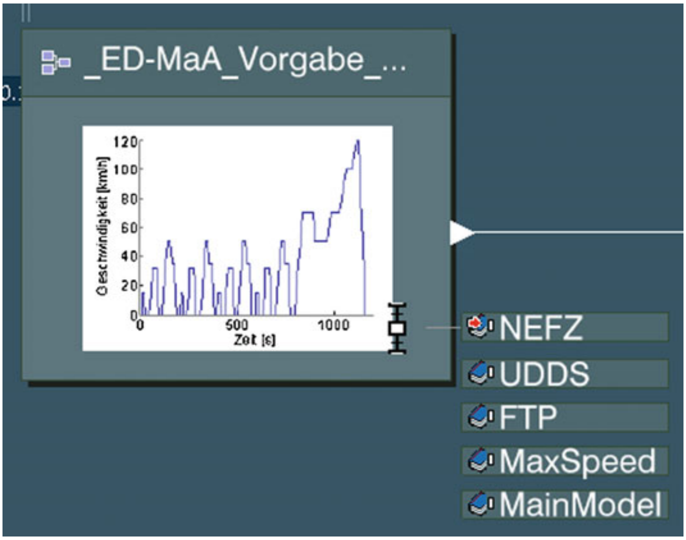


Fig. 27.6 Variation of simulation parameters

European driving cycle (NEDC), which is available on internet as value table, has been considered for this study. Other driving schedules have been considered for variation purposes (Fig. 27.6).

The driver model is represented by Block 2, which in fact is a controller that enforces the conditions prescribed by the NEDC. The controller uses the speed information of NEDC as input and triggers acceleration and braking. The controller unit of the whole system is represented in Block 3. It coordinates the different technological systems that are involved. The behavioral model of the battery module is represented by Block 4.

Since the battery has to be kept at a suitable temperature, cooling and heating are to be considered in the simulation. Block 5, therefore, represents the relevant unit.

The drive is represented by Block 6, which is an electromotor interface to the mechanical system resp. physical model through a mechanical coupling.

The chassis as well as moveable vehicle parts are building the mechanical unit, which is represented by Block 7. This unit enables taking into account dynamic driving parameters such as wind and rolling resistance.

The way of attaching models to the logical structure, for instance the mechanical model, is very important because it leads to the question, whether necessary interfaces are existing or not. Besides, this offers an opportunity to define and test standard processes. These issues are interesting from a methodological point of view.

Many simulation alternatives can be tested, whereby geometrical variations through knowledge-based engineering techniques (KBE), alternation of driving cycles as well as their values, and the change of the type of the battery cells are performed.

27.4.1.4 Selected Results

The objective of the simulation is to provide information about the performance of both battery types models (Altairnano 11 Ah, Altairnano 50 Ah) which were considered. Both battery types have been used because their modelica models already were available at one of both project partners. However, the methodology would not change if batteries of other types where to be involved in the test environment.

The results show the range of the car, the state of charge of the battery as well as the battery ageing. Furthermore, the actual speed of the simulated electric car can be compared with its nominal value. Figure 27.7 is showing such a comparison based on the NEDC.

A further point of importance in the analysis has been the thermal analysis of the cooling and heat sink. For this purpose, the thermal energy output of the battery cells has been gathered from the behavioral simulation and used as input for a transient FEA. This underlines the importance of information sharing between the different disciplines involved. Information sharing differs from information exchange in that in case of a sharing, the FEA-model can continually access necessary parameters from the battery model. When model data is just exchanged between the disciplines, mapping of both models are periodic, leading, therefore, to a lack of information on both sides and likely to data inconsistencies.

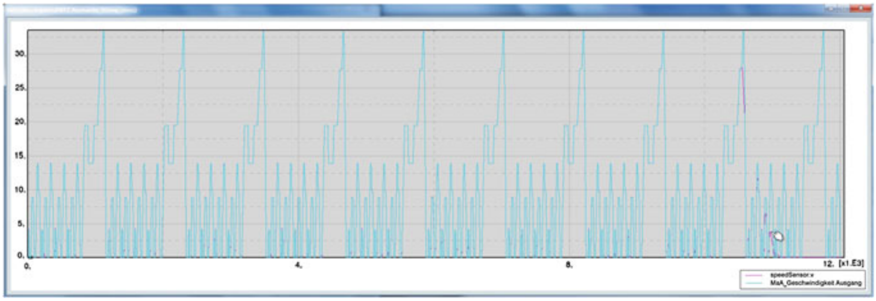


Fig. 27.7 Speed of the simulated car for a whole driving cycle according to NEDC

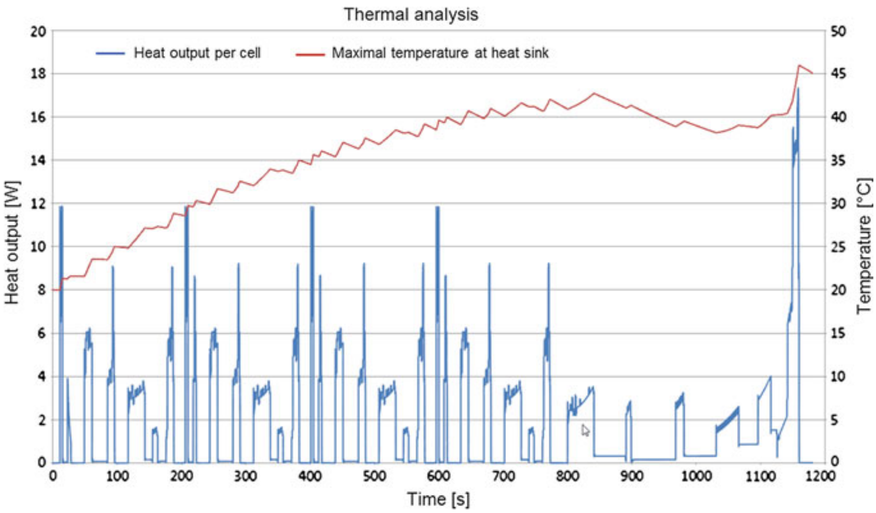


Fig. 27.8 Comparison of cell temperature with cooling temperature

Figure 27.8 shows a comparison of the cell temperature to the temperature of the cooling and heating sink. The results of the non-linear thermal analysis which has been performed are based on some assumptions made according to the project focus. For instance, the battery cells have been assembled inside a simplified, modelled battery module, that includes a simplified cooling sink. Besides the temperature distribution between the walls of the battery module have been simplified.

The case study has provided interesting information on best practices, that already is taken into account for project being implemented. The case study will be refined and enhanced in the future in order to gain more detailed information on some aspects that have been simplified, for instance the type of batteries used.

27.5 Discussion and Reflection on Concurrent Engineering

The product development following the V model (see Chap. 9), which has been partially followed for this study, provides a suitable way of working. However, applying it can be counterproductive if misused. A sequential realization of the tasks of the V model could lead to rework and, therefore, to higher costs associated with it. This could be the case if the disciplines involved work independently from each other, with the intention to merge their results at a later development stage, for instance when verifying the impact of the behavioral system on the mechanical system. For concurrent engineering to be effective, the common parameters and features that exist in the models of the different disciplines involved in product development (so-called partial models) are to be known. Relationships between these parameters are to be described in order to support a system-based model generation. Doing so would help each discipline generating its up-to-date partial model, which would have actual information from the models they are related with.

Since model or data is to be modified only by qualified personnel, the domain authority is to be respected. That is, members of each discipline should have a permission right to modify their own data (e.g., model, metadata), unless the authoring discipline explicitly has granted them modification rights.

Appropriate tools and IT services are necessary to generate domain models. Some leading solution providers have started to provide integrated environments in which different models can be mapped. However, their approach is limited to their own proprietary systems. The challenge consists of achieving a federated environment (see Chap. 4) instead of an integrated environment, in which the tools (e.g., CAD tool) and models (e.g., CAD model, behavioral model) from different vendors share data and information.

Biahmou et al. have elaborated an approach presenting the federation of the systems CATIA V5 and Matlab SimMechanics in the past [42]. In early development phases, the behavioral partial model can be derived from the geometrical one using the CAMAT (CATIA-MATLAB-Translator). A co-simulation is conducted, whereby the nominal values are sent to actuators within CATIA V5. The sensors capture values, which are sent back to the MATLAB SimMechanics model [42]. A drawback of that approach is the fact that methods must be developed to ensure the update of the different partial models involved. From this point of view, conventional autonomous tools used today and the processes based on them are not appropriate to tackle all challenges of product development. A middle way between integrated and autonomous environments is necessary (see Sect. 13.6) [43].

Based on the impact of systems engineering on all disciplines involved in product development, it might be advantageous to reorganize project teams in order to reach best performance. In fact, experience has shown that, in many companies, the requirements are shared by different disciplines, which each interpret and fulfill them, while only making little periodical adjustments together with other relevant disciplines. The overall view is, therefore, not managed since project managers not necessarily have the skills of system engineers. A promising approach consists of

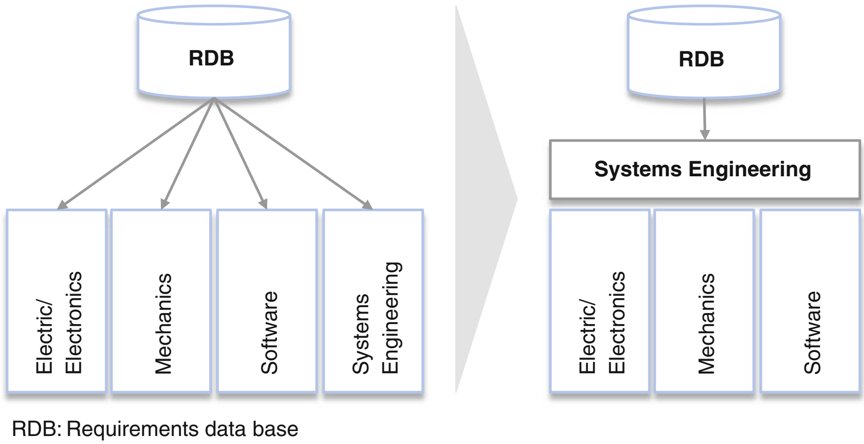


Fig. 27.9 Project organization with high performance potential

getting the requirements first in the systems engineering team, which would derive requirements for other disciplines (e.g., mechanics, electric/electronics, software), of course working in cooperation with these disciplines (see Fig. 27.9).

One of the important factors regarding the introduction of the electric vehicle is its impact on the daily life of its owner as well as on the society. Most electric vehicles have been designed with batteries to be loaded at home, even though there are some test stations for induction loading.

If the loading duration of the battery is considered, it is evident that a potential buyer would seriously ask himself what would happen the next morning, in the case he would have forget to charge the battery. This emphasizes the fact that many car makers do not seem to take seriously the fact that an electric vehicle requires new and innovative encompassing concepts rather than just introducing a battery into a car with a conventional architecture.

Electric vehicles, for instance, could be part of the power supply grid, buying and selling power according to configurations of its owner. At least, it might be interesting to equip them with the ability of automated or semi-automated triggering of the battery loading.

Besides, they could be used to stabilize the power grid which has to face instabilities due to factors such as power arising from photovoltaic solar systems or wind power plant of private persons and companies.

These instabilities nowadays already are an important challenge to be tackled by power providers. Therefore, a solution should consider the impact of some millions of batteries being charged at different times.

27.6 Conclusions and Outlook

This chapter has presented sustainable mobility as a typical field of application for Concurrent Engineering. The understanding of the term sustainability as well as its evolution in the automotive industry has been explored and actual approaches for enhancing sustainability have been presented.

First of all, the electric vehicle as an approach for alternative propulsion, but also interesting works for enhancing sustainability with alternative manufacturing processes, materials as well as drive systems have been addressed.

One important handicap of electric cars is the battery, which must power the engine. Until the customers can rely on acceptable ranges, an appropriate infrastructure and short recharging times, buying electrical vehicles certainly will remain a privilege of just a small part of the society.

Fuel cell electric vehicles are more expensive than battery electric vehicles. A fuel cell can be refilled; a battery is to be recharged. A battery stores the reluctant of its chemical reaction, fuel cell brings its reluctant to the air and, therefore, can be run as long as fuel is provided.

The focus of the work presented in this chapter has been a case study of an analysis of a battery module, a project that has been realized to contribute to sustainability through the use of an alternative, green power supply for vehicles. Since the battery design and dimensioning involves many engineering disciplines, it has been suitable for concurrent engineering application. Starting with a requirements model, a functional structure has been derived. Based on that, a logical model as well as a behavioral model have been generated. Furthermore the tools that are necessary to design and simulate an electric vehicle have been accessed during the project. It was important to emphasize the interfaces between partial models of an electric vehicle, therefore a FEA has been conducted using the physical model of the battery module.

Finally, a discussion and reflection on concurrent engineering has been made.

27.6.1 Future Perspectives

The next developments in the field of electro mobility surely will consider the question of safety when using the battery, the stability of the battery (e.g., no unintentionally battery discharge), reducing the price as well as the mass of the battery, quick loading of the batteries. Some car makers already have been facing the issue of cars burning due to battery malfunction. This calls for appropriate software, research on battery technology as well as concepts which associate power suppliers with car makers and companies, which conduct advanced research into car IT.

All these participants will have to cooperate in order to achieve synergy effects, instead of working separately, while all are pursuing the same objective. Concepts for interaction of electric vehicle and power supply infrastructures are to be developed, considering not only the battery technology, but also the availability of loading stations as well as billing concepts. Due to the fact that electric vehicles do not make the same noise as conventional cars, many people surely will not pay attention to electric vehicles driving around them. Therefore, innovative concepts for pedestrian protection will be necessary.

Stimulating the purchasing of electric vehicles remains a challenge to be tackled by politicians. It surely will be difficult to compensate the price difference between electric and conventional vehicles with only financial advantages. An additional incentive to buy electric cars may be favorable electricity prices for battery loading as well as privileges in road traffic. These may be the permission to use extra lanes, which might be otherwise reserved for special vehicles such as police cars or taxis.

Furthermore, new players could appear on the automobile market, for instance battery makers or very big IT companies such as Google, which has been conducting an important research on self-driving cars for years. This could be possible since electric vehicles do not need any combustion motor, which still represents a particular capability of today's car makers.

Sustainability will play a more important role in the future since it has become a buying criterion for cars. It is likely that potential buyers of cars will require a sustainability indicator, which takes more than the car consumption and emission into account. That indicator also would have to consider the whole lifecycle of the cars, taking the sustainability of manufacturing buildings, processes, and equipment into account [44].

Starting from the first idea of a new car to manufacturing, it will be important to assess the factors impacting sustainability in order to make an objective comparison of vehicles. New concepts of factories could be necessary to reach a high level of sustainability, for instance innovative building concepts to enable heating energy recovery.

References

1. Spath D, Pischetsrieder B (2010) Elektromobilität: Eine Technologie mit Historie und Zukunft. In: Elektromobilität—Potenziale und wissenschaftlich-technische Herausforderungen. Springer, Berlin
2. Husain I (2003) Electric and hybrid vehicles—design fundamentals. CRC Press, Boca Raton
3. Schaede H, Von Ahsen A, Rinderknecht S (2013) Electric energy storages—a method for specification, design and assessment. Int J Agile Syst Manag 6(2):142–163
4. Lorico A, Taiber J, Yanni T (2011) Inductive power transfer system integration for battery-electric vehicles. In: Hung S et al (eds) Proceedings of the 3rd international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 75–84
5. Bassett MD, Hall J, Cains T, Taylor G, Warth M, Vogler C (2012) Development of a range extended electric vehicle demonstrator. In: Proceedings of the 3rd international conference (ICSAT) on sustainable automotive technologies: driving the green agenda 2012. Woodhead Publishing, Oxford, Cambridge, Philadelphia, New Delhi, pp 205–213

6. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manag 7(2):116–131
7. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manag 6(4):307–323
8. Grober U (2009) Der Erfinder der Nachhaltigkeit. On http://www.zeit.de/1999/48/Der_Erfinder_der_Nachhaltigkeit. Accessed 15 Feb 2014
9. Schneider L, Lehmann A, Finkbeiner M (2011) Life cycle management konferenz 2011—towards life cycle sustainability management resource 4/2011
10. Peruzzini M, Germani M (2014) Design for sustainability of product-service systems. Int J Agile Syst Manag 7(3/4):206–219
11. Bruckmeier S, Wellnitz J (2011) Flexural creeping analysis of polyurethane composites produced by an innovative pultrusion process. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 13–18
12. Cannon A, Maguire M, Hulseman R, King W (2011) Manufacturing microstructured surfaces for automotive applications. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 19–24
13. Huck WR (2011) The first water based pretreatment system for direct glazing. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 25–30
14. Turner JWG, Pearson RJ, Harrison P, Marmont A, Jennings R, Verhelst S, Vancoillie J, Sileghem L, Pecqueur M, Martens K, Edwards PP (2012) Evolutionary decarbonization of transport: a contiguous roadmap to affordable mobility using sustainable organic fuels for transport. In: Sustainable vehicle technologies: driving the green agenda. Woodhead Publishing, Cambridge
15. Bunting B, Bunce M, Joyce B, Crawford R (2011) Investigation and optimization of biodiesel chemistry for HCCI combustion. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 51–58
16. Anderson CD, Anderson J (2010) Electric and hybrid cars—a history, 2nd edn. McFarland & Company, Jefferson
17. Pearson G, Leary M, Subic A, Wellnitz J (2011) Performance comparison of hydrogen fuel cell and hydrogen international combustion engine racing cars. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 86–91
18. Koike M, Miyagawa H, Suzuoki T, Ogasawara K (2012) Ammonia as a hydrogen energy carrier and its application to internal combustion engines. In: Sustainable vehicle technologies: driving the green agenda. Woodhead Publishing, Cambridge, pp 61–70
19. Dincer I (2010) Economic and environmental comparison of conventional and alternative vehicle options. In: Pistoia G (ed) Electric and hybrid vehicles—power sources, models, sustainability, infrastructure and the market. Elsevier, Amsterdam
20. Tautscher A (2012) Improving the sustainability of aluminium sheet. In: Sustainable vehicle technologies: driving the green agenda. Woodhead Publishing, Cambridge
21. Bansemir H (2011) Design of basic structural composite elements: In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 95–102
22. Hufenbach W, Krahl W, Kupfer R, Rothenberg S, Weber T, Lucas P (2011) Enhancing sustainability through the targeted use of synergy effects between material configuration, process development and lightweight design at the example of a composite seat shell. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 103–110
23. Kajtaz M (2011) Sustainable design of a side door reinforcing assembly—exploratory optimisation. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 111–120

24. Müller L, Wellnitz J (2011) Research and development of a new and sustainable composite: "natural stone laminate". In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 121–127
25. Hofmann P (2010) Hybridfahrzeuge—Ein alternatives Antriebskonzept für die Zukunft. Springer, Wien
26. Wilhelm E, Wellnitz J (2011) RPM—Rotatory piston machines: new class of innovative machines. The first water based pretreatment system for direct glazing. In: Hung S et al (eds) Proceedings of the third international conference (ICSAT) on sustainable automotive technologies 2011. Springer, Berlin, pp 67–71
27. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manag 7(2):101–115
28. Weber J (2009) Automotive development processes. Springer, Berlin
29. Biahmou A (2005) Methoden für das Industriedesign in Virtueller Realität. PhD thesis, University of Technology, Berlin
30. Dineva E, Bachmann A, Moerland E, Nagel B, Gollnick V (2014) New methodology to explore the role of visualisation in aircraft design tasks: an empirical study. Int J Agile Syst Manag 7(3/4):220–241
31. Dhameja S (2002) Electric vehicle battery systems. Newnes, Boston
32. Peng et al (2013) Development of an open architecture electric vehicle. In: Azevedo A (ed) Advances in sustainable and competitive manufacturing systems. Springer International Publishing, Switzerland
33. Stark R et al (2010) Cross-domain dependency modelling—how to achieve consistent system models with tool support. In: Proceedings of 7th European systems engineering conference, EuSEC 2010
34. Josuttis N (2008) SOA in der praxis: system-design für verteilte Geschäftsprozesse. Dpunkt Verlag
35. Prostep iViP Association. http://www.prostep.org/de/cpo.html. Accessed 15 Feb 2014
36. VDI-Gesellschaft. About us, http://www.vdi.eu/about-us/. Accessed on 15 Feb 2014
37. Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Engineering design. a systematic approach, 3rd edn. Springer, Berlin
38. Biahmou A (2012) An efficient CAD methodology for glove box design. In: Stjepandić J, Rock G, Bil C (eds) Concurrent engineering approaches for sustainable product development in a multi-disciplinary environment. Proceedings of the 19th ISPE international conference on concurrent engineering. Springer, London, pp 219–229, 2013
39. VDI-Gesellschaft Entwicklung Konstruktion Vertrieb (2004) VDI 2206—design methodology for mechatronic systems. Beuth Verlag, Berlin
40. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3/4):324–347
41. Fritzson P (2011) Introduction to modeling and simulation of technical and physical systems with modelica. Wiley, Hoboken
42. Biahmou A, Fröhlich A, Stjepandić J (2010) Improving interoperability in mechatronic product development. In: Proceedings of PLM10 international conference, inderscience, Geneve
43. Fukuda S, Lulić Z, Stjepandić J (2013) FDMU—functional spatial experience beyond DMU? In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 431–440
44. Mo J (2014) Assessing the requirements and viability of distributed electric vehicle supply. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 74–83

Part IV
Current Challenges

Chapter 28
Challenges of CE

Wim J.C. Verhagen, Josip Stjepandić and Nel Wognum

Abstract Despite a long pedigree and many positive reports on its use and benefits, concurrent engineering (CE) and its associated research (sub)domains still experience significant development. In this final chapter, a socio-technical framework is applied to classify and analyze challenges identified as part of the foundations, methods and applications discussed in this book. Existing properties and means of CE are abstracted. Subsequently, the main trends and developments in CE research and practice are discussed, followed by expectations for the future. Findings and trends have been identified for strategic issues visible in product requirements and product portfolios, stakeholders including companies involved, multiple functions and disciplines, current and future technologies that are expected to solve at least some of the existing problems, knowledge and skills as brought by people and teams, and structures necessary for making collaboration work, while dealing also with the still very difficult cultural differences. As the chapter shows, CE as a concept is very much alive, requiring even more advanced tools, techniques and methods to contribute to less waste in resources and efforts world-wide and improve quality.

W.J.C. Verhagen (✉)
Aerospace Engineering, Air Transport and Operations, TU Delft, 2600 AC Delft, The Netherlands
e-mail: w.j.c.verhagen@tudelft.nl

J. Stjepandić
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: josip.stjepandic@prostep.com

N. Wognum
Social Sciences Department, Management Studies Group, Wageningen University, Hollandseweg 1, 6706 KN Wageningen, The Netherlands
e-mail: wognumnel@gmail.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_28

28.1 Introduction

The foundations, new developments, methods of Concurrent Engineering (CE) and its applications in several industries have been discussed at length in the preceding chapters of this book. As part of the respective chapters, authors have discussed specific research and practical challenges in isolation. The aim of this final chapter is to integrate and present current challenges with respect to CE and its associated research (sub)domains in a comprehensive overview. In order to perform this integration in a structured, methodical manner, a framework covering the socio-technical dimensions of CE has been devised. Figure 28.1 shows this framework.

The framework is intended to capture the full breadth of the state of the art and developments in CE, and as such considers the socio-technical dimensions of the CE system as introduced in Chap. 2. To recapitulate, these dimensions are:

- **Strategy/goals**: overall goal or vision of the CE system, which may include aspects such as profitability, sustainability, availability, efficiency, etc. Strategy and goals become visible in a product/project portfolio or ideas and requirements for a new (package of) product(s) and the associated processes.
- **Stakeholders**: actors, functions, disciplines or roles that are involved in the process(es) composing the CE system. Not only directly involved stakeholders (internal organisation, supply chain partners) may be considered, but also external stakeholders that influence the process but have no direct involvement (e.g., governmental institutions, technology providers).
- **Technology**: the technologies required to control and manage the processes associated with the CE system. This includes hardware as well as software, in particular basic technology (e.g., internet).
- **Knowledge/information**: the knowledge or information required to initiate and sustain the processes composing the CE system. This comprises the tacit and explicit knowledge available in people as well as information systems. It may include internal and/or external information sources.
- **Organisation/structure**: the configuration of the processes required to run the CE system. This may include aspects such as location, hierarchy, teams, communication lines, degree of outsourcing, and structure, culture, and so on.

Additionally, the proposed framework (as given in Fig. 28.1) incorporates another axis of analysis, which considers the current and future state of the CE system dimensions. To be more specific, each dimension is considered in terms of its:

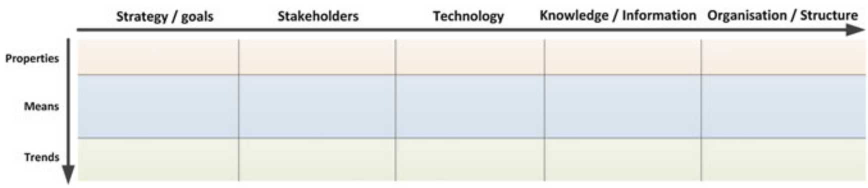


Fig. 28.1 Framework for analysis and representation of CE challenges

- **Properties**: which essential characteristics of the CE system dimensions and existing solutions can be discerned?
- **Means**: how are the properties actualized in current engineering practice, in particular in the main industries? Which methods, techniques and tools are available to support application?
- **Trends**: which developments can be seen with respect to the properties and means of the CE system dimensions? Which changes and/or evolution can be observed, anticipated, and expected? Is there, maybe, a possibility for disruptive change? Could CE become obsolete (e.g., through new technologies and approaches)? Which new fields can be addressed (e.g., the rising area of crowd sourcing)?

Through application of the framework on the individual elements of CE as discussed in this book, a comprehensive overview of challenges in CE can be generated. As such, the chapter structure reflects the main elements of the framework. In Sect. 28.2, the properties of CE solutions are discussed. Following this, Sect. 28.3 incorporates analysis of current CE tools and methods that are available to support application. In Sect. 28.4, the major contribution of this chapter, trends in research and practice are highlighted and discussed in detail. In Sect. 28.5, expectations with respect to further development of CE related to the major recent technological trends are explained. Finally, a summary of findings is given.

28.2 Properties of CE Solutions

The first step in applying the analysis framework is to analyze which essential characteristics of the CE system dimensions and existing solutions can be discerned. These generic properties of CE have contributed to its current achievements and point towards existing challenges, as will be further discussed in Sect. 28.4.

28.2.1 Strategies/Goals

Current properties of the CE system in relation to its goals and associated strategy are strongly associated with the multidisciplinary, product development oriented nature of the CE philosophy. Many authors speak about trans-disciplinarity in this regard (see Chap. 4) [1]. Furthermore, a network-centric approach can be discerned [2].

As part of new product development (NPD), CE comprises coherent methods and tools for improving performance of a product development process. Important strategic properties in this respect have been and still are the expected reduction in time and costs and improvement of quality of the NPD process and outcomes. It is very important for enterprise to shorten their time-to-market to stay ahead of competition. In recent years, three interrelated strategic properties of CE systems have been strongly emphasized:

- **Integration**: one of the hallmark properties of CE is its focus on integration and simultaneous consideration of lifecycle requirements and constraints in the NPD process. Alongside process integration, supporting functions are increasingly integrated as well. The most obvious examples are organizations and information systems [3]. Organizations (as discussed at more length in Sect. 28.2.5) not only wish to integrate functions within the internal organization, but also endeavor to involve suppliers, logistic partners and customers/end users [4]. With respect to information systems, organizations have sought to connect and integrate their data, information and knowledge both internally and externally, for instance through product data management (PDM) and product lifecycle management (PLM) technology [5]. More recent developments have focused on cloud storage and computing, and Software-as-a-Service (SaaS). In support, Chap. 4 (Technology Foundations) has highlighted the need for a service-oriented programming methodology to accompany a common design process, domain-independent representations of designs, and general criteria for decision-making. Chapter 6 (Resolving Interoperability in CE) has discovered the corresponding methodological solutions. Seamless integration has become a pre-requisite for trans-disciplinarity (see further Sect. 28.4.5) [1, 6–8].
- **Complexity**: from a strategic perspective, organizations have to cope with massive complexity from internal and external sources [9, 10]. As highlighted in multiple instances in this book, competitive pressures and increasing customer and supply-chain involvement make the NPD process more complex, for instance through the need to resolve larger sets of sometimes conflicting requirements and constraints [11, 12]. Complexity acts in a self-intensifying manner: many new product variants are designed because developers were not able to select and re-use an existing one due to the sheer number of available options. In many cases, the number of possible product variants is much higher than the number of sold products, as highlighted in Chap. 17 [13, 14]. Internally, the drive for integration can clash with the legacy organization and technology, leading to complex arrangements. The presence of countervailing forces driving product and portfolio complexity points to the need for developing strategies that can yield optimal levels of complexity like standardization and modularization that in turn result in maximal performance of products, portfolios, and supply chains [10]. The approach to manage complexity in design or manufacturing varies from coping with it, and trying to manage it to advantage, to minimizing or eliminating it, as explained in Chap. 14. Given the huge challenges facing engineering, which are of increasing complexity in breadth and depth, it is realized that companies must consider complexity in technical as well as in other multi-disciplinary domains [9, 15]. Thus, manufacturing enterprises will need to not only to adopt flexible technical solutions, but they must also effectively innovate and manage complex socio-technical systems [16] (see Chap. 8).
- **Stability and resilience**: increasing integration and complexity threaten the stability of products and processes in NPD. For instance, as highlighted in Chap. 5 (Requirements Engineering), requirements may present significant instability as major shifts in requirements can occur during early design,

prompted by the interacting influences of multiple stakeholders. From an application perspective, additional requirements posed during a product lifecycle (e.g. stricter sustainability requirements) may threaten the stability of a product-service system [17]. In the face of instability, the strategic capacity of an organization to respond quickly and effectively to instability, as incorporated in the concept of resilience, assumes importance as a strategic rather than operational property [10]. Furthermore, agility remains an important capacity to react almost spontaneously to any market change and set-up, run, and terminate quickly a co-operation as well [18].

28.2.2 Stakeholders

As already intimated in the previous subsection, an important characteristic of CE is the inclusion of multiple lifecycle phases, disciplines and domains into a single consistent design process. As such, CE stakeholders range across the entire value chain and are united in the following stakeholder properties:

- **Multidisciplinary processes**: NPD according to the CE philosophy involves the (simultaneous) incorporation of stakeholders' requirements, constraints and contributions in the design process. From an internal perspective, organisations bring together the required disciplines for NPD into integrated teams, supported by appropriate tools and methods [19–21]. Part of this process is becoming automated (in a repeatable fashion) through techniques such as knowledge-based engineering (KBE) and multi-disciplinary design optimization (MDO), which can be used to reduce design time while ensuring optimization for multidisciplinary criteria [22]. From an external perspective, customer/end user involvement is increasing, before and during the NPD process as well as in the product service phase [17, 23].
- **Socio-technical systems**: CE considers the product development process as a socio-technical process. The interaction between technical and social characteristics is seen as a driver for problems that emerge in complex engineering programs. Current complex projects with actors from different disciplines and culture display dynamics that cannot be fully anticipated as of now [24]. Dealing with such systems under rising complexity requires technical as well as social/organizational methods and tools [25, 26].

28.2.3 Technology

To enable the application of CE in practice, technology plays an indispensable role. Achieving concurrent design of products in multi-site, multi-organisational and multi-disciplinary environments requires strong support of (information) technology [27]. A wide range of methods, tools and applications is available (as further discussed in Sect. 28.3.3).

In terms of properties of technology in association with CE, a number of critical properties can be discerned. First of all, *accessibility* is critical: supporting tools and applications should be accessible anywhere, anytime. Anything we do must become immediately transparent for the authorized stakeholders' circle (internal and external) to facilitate accessibility of information. In response, the technology foundations of CE are increasingly moving towards service-oriented concepts and trans-disciplinarity. Service orientation simplifies the provision of the up-to-date information. Finally, optimal accessibility preserves reliable processes to ensure that each input induces the output expected.

Second, *interoperability* of tools and applications is critical to facilitate collaborative engineering. The main processes must be harmonized, in particular in global contexts, to mitigate cross-cultural differences. In particular, systems that support negotiation and information exchange are required. Collaboration is deemed essential for resolving conflicts, handing workload balance, and reducing development lead time and costs. Interoperability requires common infrastructures and corresponding standards [28].

Ease of use and *maintainability* are further technological properties to consider. The previously mentioned dramatic increase in complexity in product development also acts on technology, as interoperability of increasing numbers of systems and applications is required. In the face of this development, end users must retain the capability to work efficiently with the provided technology. Provision of easy-to-use common clients for a plethora of applications remains still a challenge with huge potential benefits, as highlighted in Sect. 21.4.1. Maintenance of interwoven, complex systems is likely to drive up costs, an unintended and to be avoided effect of increasing interoperability.

Finally, the right level of *fidelity* should be available during the design process. As the design process itself is shortened in time, the level of detail is shifted forward as well to speed up design as well as to enable more detailed, accurate analysis. Technologies such as CAD, KBE, PLM, DMU, etc., are increasingly used in early design.

28.2.4 Knowledge/Information

To support the simultaneous and collaborative elements of CE, a number of properties related to information and knowledge are critical to achieve [29].

First of all, *exchange* of information and knowledge is crucial to establish. For instance, early phases of design are frequently characterized by a large amount of very different trade-offs between concepts answering customer and functional requirements, as highlighted in Sects. 22.3.6 and 23.2.2 [26, 30, 31]. To adequately come up with and select the right concept(s) for further development, knowledge of design as well as the implications of design choices is required. Exchange of information and knowledge between stakeholders is paramount to achieve this deep understanding. At later stages of design, cost, time and quality of design may be favorably impacted by information and knowledge exchange [32].

Knowledge retention and re-use is another important property in the light of CE. From one design iteration to the next the knowledge underlying the design process must be retained and, if possible given the context, re-used. Reinventing the wheel is avoided; on the contrary, the design process may be sped up through efficient application and re-use of existing knowledge. In a more extreme form, this principle underlies the automation of design processes through KBE solutions [33] and modularization of products and processes as well, as described in Sect. 14.5.3. They can improve product quality by re-use of well-proven solutions.

Human involvement in information and knowledge generation, exchange and (re-)use is characteristic of the socio-technical perspective that is embodied in CE. Humans are an indispensable part of knowledge processes; their generation, interpretation and use of knowledge during the design process will generally drive the direction of the process. In various applications of CE, as well as related research domains, humans are explicitly or implicitly included as part of models, methods, tools and applications. Team-oriented education in a realistic environment for CE, addressing methodological, technological, and social skills, plays an important role [34].

Finally, *ownership* is a crucial property of information and knowledge. Chapter 18 (Intellectual Property (IP) Protection) is fully devoted to this issue. The increasing degree of collaboration in global relationships (e.g. supply-chain partnerships), ubiquitous digital communication techniques as well as tough competition necessitate careful consideration of ownership and protection of knowledge. Resolving this issue would help to keep and increase acceptance of information and communication (ICT) technology [35].

28.2.5 Organisation/Structure

The *multidisciplinary* nature of CE has already been emphasized. By definition, this is an important property of any CE system. Inclusion of multiple disciplines in the NPD process will increase 'first-time right' development, while better anticipating and acting on lifecycle requirements, leading to lower overall lifecycle cost and better quality.

An important organizational property in relation to CE is its actualization: how is the organization arranged around the CE-compliant design process? The organization of CE processes is often made visible in the team structures chosen. Main questions are whether teams need to be co-located or not and which additional measures need to be taken to make teams really working together, taking into account the cultural differences that team members bring to the team (see e.g., Chap. 8). The means to achieve a smooth organization are further touched upon in Sects. 22.3.6 and 28.3.5. The need for proper composition of design teams has also been highlighted in many of the application chapters of this book, for instance in aviation, machinery and medical applications (Chaps. 20, 22 and 25).

Contracts between stakeholders are another issue of organization. They may include safeguards against knowledge leakage or misuse as discussed in Chap. 18. In case of repeatable, long lasting partnerships the methodology to introduce, maintain and discontinue a partnership need to be elaborated on an industry level, as highlighted in Sect. 21.5.

A final property of interest is the level of *standardization* associated with products and processes. Standardization may help to achieve desirable properties that have been mentioned before, such as interoperability (Chap. 6). However, the effort to standardize comes with its own costs, and may impact the capability of an organization to respond to localized (i.e., non-standard) issues. Good examples are given by the global automotive and aerospace industries, where many standards have been developed and adopted to facilitate partial processes of CE with participation of IT vendors (see Sects. 6.5.1 and 21.6). The "Code of PLM Openness" (CPO) is a further initiative, which will facilitate global progress in standardization [36].

28.3 Means: Current CE Tools and Methods

The second step in applying the analysis framework is to analyze which means can be discerned to bring into practice the CE principles and properties. This section consequently gives an overview of current tools and methods and their application areas according to the framework, including scope, limitations, and identified obstacles.

28.3.1 Strategies/Goals

The means available to operationalize the strategic properties of CE are numerous. With respect to integration, complexity and resilience, methods and tools for interoperability and SaaS can be used to achieve the desired properties.

The means for achieving interoperability are covered at length in Chap. 6. Standards have been defined (e.g. ISO STEP) and supporting languages are available, e.g., semantic-oriented service languages (RDF, RDFS, OWL), and service description languages [Web Service Modelling Language (WSML), Unified Service Description Language (USDL)]. The Service Oriented Architecture paradigm with its associated concepts of loose coupling, services and interoperability can be adopted to achieve interoperability. Supporting methods include variants of semantic mediation. Additionally, frameworks and models are available to describe and guide interoperability development, including the ATHENA framework and the Systems-of-Systems Interoperability (SOSI) model, which have been subsumed into federated form. Interoperability is closely related to developments in offering SaaS. Recent work on a service-oriented programming methodology has been highlighted as part of Chap. 4: Technology Foundations. The deployment of standardized lightweight formats like JT and 3DPDF, as highlighted in Chap. 11 and Sect. 23.3.2, can significantly simplify the collaboration among internal and external partners under protection of the IP. Meanwhile ISO STEP is under continuous maintenance: AP242 will serve as umbrella for other standards (e.g. JT).

28.3.2 Stakeholders

The means to increase stakeholder involvement in NPD include company-internal as well as external solutions. Early customer involvement remains crucial [23] and presupposes appropriate requirements management, as highlighted in Chap. 5.

From the internal perspective, solutions such as PDM, PLM and MDO allow for the simultaneous inclusion of engineering function stakeholders into the overall process. From an external perspective, supplier involvement is covered by PLM solutions as well as by dedicated supplier portals. An example covered in this book is the Boost Aerospace portal used for supplier information exchange in the aviation domain, as highlighted in Sects. 6.5.1 and 20.5.1.1. At the other side of the value chain, customers can increasingly be involved in the NPD process through direct and indirect means [23]. Direct means include the use of social media and focus groups. Indirect means include data analytics of customer behavior and market trends.

28.3.3 Technology

The technology means for actualization of CE properties are very numerous; multiple methods, models and tools exist; sometimes complementary, sometimes in competition. In fact, much of the content in Section II of this book focuses on methods, tools and applications of technology means. Giving an extended overview of this content would be repetitive and unnecessary. Instead, Fig. 28.2 maps the covered domains to a common representation of the systems engineering (SE) process. What is missing in the figure, are the set of tools, methods and techniques for structuring and managing teams, and to support collaboration and communication in these teams, whether co-located or not. Actually, team performance is important in all phases of product and process development. The requirements for team formation and management, however, vary with the different phases of the NPD cycle as highlighted in several chapters (e.g., Chaps. 8 and 22).

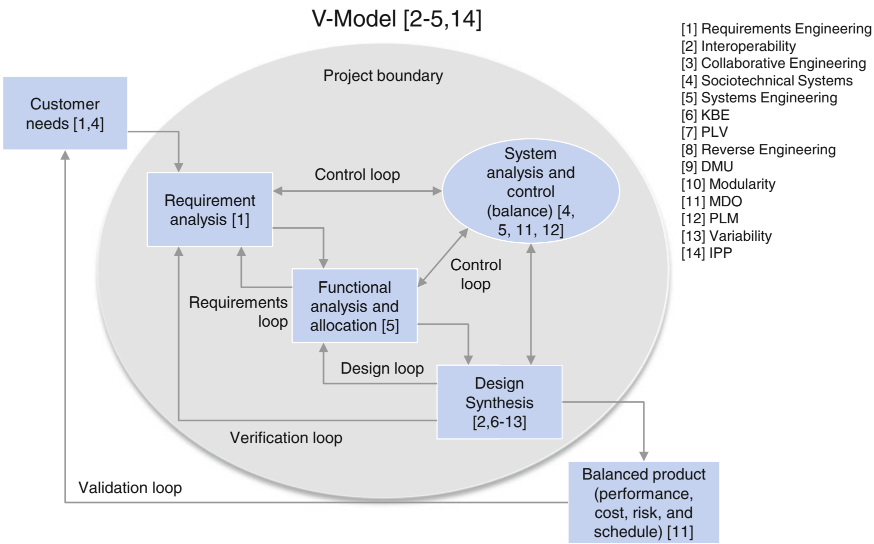


Fig. 28.2 Research domains and associated means mapped to systems engineering V-model

28.3.4 Knowledge/Information

In today's (concurrent) engineering practice, huge amounts of data, information and knowledge are being generated and exchanged on a daily basis. Means to capture, store, exchange, use, retain and maintain this volume are an integral part of the CE approach.

Capturing as well as using data, information and knowledge is often an implicit part of using engineering tools (e.g. CAD suites) to generate product solutions. PDM and PLM systems explicitly emphasize data and knowledge capture and (re) use. At the 'extreme' end, KBE solutions require explicit capture and formalization of knowledge before being able to automate engineering processes. Besides a technical element, knowledge capture has a strong social element: the willingness of experts to share knowledge as well as the ability to elicit knowledge properly and the ability to offer correct and valid knowledge strongly involve social processes.

For storage, retention and maintenance of data, a multitude of technological solutions is available, ranging from relatively straightforward office applications to dedicated databases to value-chain spanning PDM systems. Information storage, retention and maintenance involve connecting data using contextualization and semantic annotation. Knowledge storage, retention and maintenance may involve the use of dedicated knowledge bases, where engineering rules are stored in anticipation of further use with facts and inference engines. Knowledge management solutions may facilitate the interaction with knowledge bases. Retention involves a social component as well, in having methods and processes to keep knowledge available and current, vis-à-vis reinventing the wheel.

The exchange of data, information and knowledge is supported by a wide range of methods and standards, as detailed in Chap. 6. The most well-known and widely used standard in this regard is ISO 10303 Standard for the Exchange of Product model data (STEP), with its various application protocols (APs), which is maintained and updated continuously.

Insufficient data quality is identified [37] as a major obstacle for proper re-use of information and knowledge, in particular in case of translation and migration. While generating product data with a certain application (e.g. CAD), it is difficult and sometimes impossible to anticipate the requirements of all involved stakeholders to this data in later stages of the product creation process. Especially when shifting to the next generation of software, massive issues with old data can be expected due to the change in data model(s) and underlying consistency criteria. Subsequently, product data quality must be subject to continuous monitoring and improvement to avoid the need for rework and remastering of the original data. In contrast, good data quality can be ensured by well-trained staff using comprehensive methods in a stable IT environment.

28.3.5 Organisation/Structure

Various methods and models are available to support the organization of multidisciplinary teams relative to NPD. Integrated Product Development Teams (IPDT) and Design-Build Teams (DBT) are examples of team organization modes, which incorporate experts from multiple lifecycle phases to represent disciplinary requirements, constraints and solutions.

IPDT and DBT are outflows of the realization that humans need to be 'in the loop', especially in the earlier phases of design. Despite available precepts, organization and management of multi-disciplinary teams in these phases is still a challenge. Who needs to be involved? Which stakeholders? How to create consensus and commitment? Often, cultural differences are not sufficiently well understood. These issues are further discussed in the following section: Trends in research and practice.

28.4 Trends in Research and Practice

In this section, trends in research and practice for the (near) future are identified. This analysis is motivated by the existing awareness that CE is still very much alive. Much effort is required to achieve a concurrent way of working in current complex projects, supply chains, and networks that focus on development of new products and processes.

28.4.1 Strategies/Goals

From a strategic perspective, numerous trends in CE can be discerned. The main trends are summarized below.

First of all, product development is an increasingly global activity. This poses significant challenges on interoperability of method, tools as well as organizations and users. In relation with this, the dynamic interaction between technical and social characteristics of new product (and service) development needs to be taken into account in models, methods and tools. Because prediction of emergent behavior, due to this dynamic interaction, is currently not supported well by existing CE means, improvement in models, methods, and tools is needed. In particular, given the mentioned trend towards globalized product development, cultural differences need to be modelled and incorporated into methods.

Globalization as well as increasing complexity are drivers of increasing integration of method and tools. Integration and interoperability are assumed to speed up development and lower costs, yet developing and implementing interoperable, integrated solutions can be assumed as a driver of complexity as well. To simplify these aspects, existing standards may be employed in support. Further efforts to standardize aspects of integration and interoperability are underway (see Chap. 6). Besides standardization, there is a trend towards loosely coupled models in federated environments. On a local level, users can specify their domain models without worrying about integration aspects. On a global level, the federated framework takes care of model integration and interoperability. Furthermore, globalization requires a high level of time synchronization of distributed teams.

Another major strategic shift is servitization of manufacturing industries, i.e., the innovation of organization's capabilities and processes to shift from selling products to selling integrated products and services that deliver added value. This has already started, as illustrated by initiatives such as BoostAerospace, or the role of Cloud solutions for setting up the infrastructure of the IMAGINE project (see Chap. 6). Servitization provides means for companies to move up the value chain and exploit higher value business activities. A service-led competitive strategy sees everything-as-a-service where Cloud Computing is seen as a major trend. From a practical perspective, two directions for its further development can be observed [17]. First, depending on the industrial sector, manifold options and barriers must be considered and broken down to develop a marketable service offering. Second, traditional manufacturers will have to cope with the challenges of adopting a novel business model while retaining internal and supporting activities.

In this context, the field of service engineering enables us to innovate, design, and manage simple as well as complex service operations and processes of the intelligent service-based economy [4]. Following current trends, increased integration of digital components in product-service systems and increased relevance of digital context in servitization are crucial constituents of such offerings. Information and communication technology serves not only for asset monitoring (use, condition, location), but for operational (monitoring and cost control) and strategic roles (feedback to customer and research and development) as well. “Digital Servitization” defines the provision of IT-enabled (i.e., digital) services by relying on digital components embedded in physical products [3].

Many aspects of servitization have already been implemented in a piecemeal fashion in various Maintenance, Repair and Overhaul (MRO) industries (e.g., diagnostic systems, condition monitoring), but integrated, fully digital solutions are often missing. To get there, the MRO industry will at first have to transit from a paper-based to a paperless organisation. Moreover, supporting tools will have to leverage data availability by including functionality for diagnostic, prognostic and (optimized) planning and execution purposes [38]. With respect to the latter issue (maintenance execution) as well as digitalization of paper processes, the advent of mobile tools (the use of smartphones, tablets, augmented and virtual reality solutions in combination with web-based or bespoke information systems) and embedded devices holds significant promise in making digitalization and execution more efficient.

28.4.2 Stakeholders

The aforementioned growth in complexity and increasing integration and interoperability will have a significant effect on the stakeholder environment, a process that has already been ongoing. With increasing complexity and integration, the size and complexity of the stakeholder environment are also expanding.

First of all, this has implications for short-term dynamics in stakeholder environment composition. As noted in Chap. 5, the realization of complex systems usually requires the temporary collaboration of a multitude of stakeholders from different domains, such as hardware, software and services. Besides the customer/ user and the system integrator, there are stakeholder groups for the system components, life cycle services and system environment, each with their own objectives and context [39]. During the various stages of the design process, these stakeholders will generate dynamic and sometimes conflicting sets of requirements [40]. Involving all stakeholders continually in the process may very well drive up overall design time. To counter this, techniques such as agile design and development, where fast prototyping, test-driven, model-driven and behavior-driven development methodologies allow focusing on specific business cases, may be employed.

The stakeholder environment is also subject to long-term dynamics. In this light, the previously mentioned trend towards servitization will impact stakeholder composition. In the digital context, organizations are increasingly focusing on value creation outside their boundaries, because the value is created through interplay of customers, competitors, collaborators and the wider community. In terms of product lifecycle management, this trend gives after-sales equal importance as other phases of the product lifecycle, including added value generated from Big Data and Internet of Things (see Sect. 28.5). Depending on the exploitation of the product (e.g., a passenger car), a vendor can build a user profile of the customer and, based on this, offer additional services (route optimizing, payment, infotainment, maintenance) using the recorded operating data. In the sum, the initial product would become (more) "intelligent" and give additional value to the customer [41, 42]. Security issues and privacy concerns remain obstacles to be dealt with.

Likewise, the call for inclusion of socio-technical modelling in CE will involve bringing in stakeholders to represent the social and cultural elements of design. The interaction between designers and product users as well as other stakeholders must be modelled using insights and results from the social and human sciences.

28.4.3 Technology

Trends in technology to support CE are closely related to the strategic trends that have been identified.

With respect to collaboration and interoperability, new interoperability requirements have to be identified, which describe organizational, technical and management prerequisites for the system realization. New concepts and techniques will have to support two main aspects:

- Collaboration and interoperability between stakeholders and system components from different domains, especially hardware, software and services.
- Improvement of singular processes (e.g. management of unstable and unknowable requirements), tools and applications.

Four main fields of action for the latter aspect are depicted in Fig. 28.3: development processes and methods (1), customer processes (2), downstream

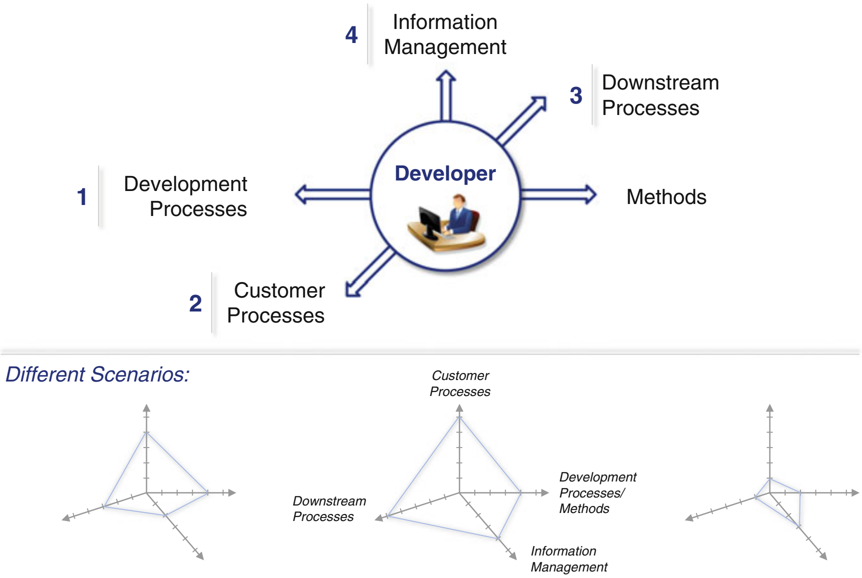


Fig. 28.3 Fields of action for improvements of CE

processes (3), and information management (4). In this figure three typical scenarios are shown in corresponding enterprises with respect to how to deal with such challenges. These fields of action can be considered independently, but the integral approach in multiple steps promises higher benefit. The order of steps must be defined carefully and depends on the specific situation.

Usually, the improvement begins with development processes, which are supported through corresponding methods (Requirements Engineering, SE, KBE, MDO, etc.) (number 1 in Fig. 28.3). It's easy to implement new methods because the count of users is (relatively) limited, the degree of specialization is high, and, thus, the risk is low. Enterprises which invest here (keyword: Idea-to-Order) harness the benefit through a fast, lean, and efficient development process, which rapidly transforms development plans into products. This field discovers dynamic research and development of new tools for each specific application. Leading vendors like Dassault Systemes, PTC, and Siemens PLM offer ever-new modules of their standard products like CATIA, Creo, and NX. Such improvements can be verified rapidly and bring quick wins due to process acceleration.

For suppliers in particular customer processes are crucial (number 2 in Fig. 28.3). In such a way pre-requisites are fulfilled for winning certain customer's order. Collaborative Engineering is the first pre-requisite (see Chap. 6) supported by corresponding methods and tools. It is easy for a supplier to exaggerate when one "IT island", which comprises customer-specific tool and methods, is built for each customer. Such islands cause subsequent efforts in the downstream processes (number 3 in Fig. 28.3) and in information management (number 4 in Fig. 28.3). It looks like segmentation of the entire enterprise. Such constellations can be discerned in Fig. 28.3 below left where information management faces huge challenges from management of such islands. In this case a supplier is seeking tools and applications for harmonization of requirements, unification of singular steps in the customer process, customer data exchange, IP protection, etc.

For most enterprises the highest priority lays on the improvement of internal downstream processes, because such measure affects many internal users and can serve as process backbone. Assuming proper implementation of downstream processes, it is easier to adjust development processes and information management, respectively. This is the context that heavily enforces the development of lightweight 3D formats (JT, 3DPDF) and digital mock-up as discussed in Chaps. 11 and 13, respectively. The ever-attractive idea of one format for all users certainly will justify extensive harmonization and standardization activities in the future. For customer processes either standard interfaces are used or just service providers who adapt the data for exchange while keeping costs low (Fig. 28.3 below mid). Actually each large company deploys its own supplier portal based on or closely integrated with its PDM system. There is a need, not a trend yet, for a standardized portal like the solution presented in Sect. 23.3.2.5.

Finally, there are some enterprises that have exceptionally high confidence in information technology. They introduce modern CAD, PDM and various simulation systems and often neglect the optimization of processes and customization of software tools. High transparency of data is easily achieved by PDM, but their processes are static and don't reap expected benefit (Fig. 28.3 right). Such enterprises must learn that technology alone is not enough and must subsequently invest more in process definition, harmonization, development of methods, and user training.

If all four aforementioned action fields are implemented properly, a virtual product can be created and corresponding processes can be operated. Further improvement in technology concerns the growth in importance of virtual products (see Chaps. 13 and 15). A long-term trend can be discerned, which will be enforced by new techniques like Functional Digital Mock-Up (FDMU—see Sect. 13.6); from an idealistic point of view, no real (physical) product would exist until the start of production. Virtual product presentation would be the sole representation; physical prototypes would not be constructed. However, even allowing for this scenario, at least two use cases still show the need for seamless coexistence between virtual and real products for success of CE.

First of all, for enhancement, functional extension, overhaul, or just refurbishment of existing products (see Sect. 13.7.2) it is necessary to superpose the virtual parts with the existing real parts of emerging products. That can be conducted by using Virtual and Mixed Reality (VR + MR) techniques. One can, therefore, easily imagine that such approaches could be used to extend and validate platform products. As shown in Fig. 28.4, MR faces a very promising future in most phases of the product creation process of a passenger car [43]. Subsequently, supply chain visibility of a product is needed for tracking the product [44].

Secondly, rapid prototyping technologies may experience significant growth by the broad introduction of desktop 3D printing technology. The term "built or produced on demand" will obtain a new meaning. The rising variety of used materials with improved technical properties opens entirely new options for rapid and low-cost production of spare parts. Last but not least, a closed-loop in the mutual interoperability between virtual and physical products using rapid prototyping and reverse

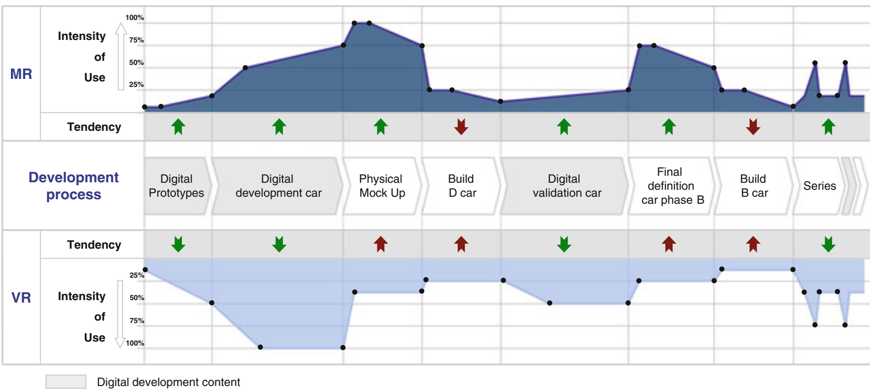


Fig. 28.4 Long-term forecast for use of virtual and mixed reality at daimler (reproduced with kind permission from Daimler AG © 2014 [43])

engineering techniques (see Chap. 12) is still needed in the product creation process of many products [45], in particular in case of reconstruction of parts of the human body (e.g., dental applications, prosthesis) [46] or when the shape of the product is defined by a stylist.

28.4.4 Knowledge/Information

The aforementioned trends towards integration and interoperability have an impact on the exchange of knowledge and information. Using technology means, sharing has become easier than ever. The amount of data being created, stored and used every day is growing exponentially. The way in which the knowledge is used in the design process is changing continuously. Some of the driving factors are listed below [47]:

1. **The rise of the wikis**. A wiki is a database of interactive web pages that allows members of a user group to collectively edit the same material from any computer with an Internet connection. Wikis provide a flexible and self-organizing platform that is especially useful from the point of view of early design, when the information and knowledge is unstructured, and from the point of view of collaborative design, where all communication is persistently recorded and loosely organized through user-defined tags. With such capabilities wikis aims to fill gaps left through large software systems in almost each enterprise.
2. **Bio-inspired knowledge for design**. Bio-inspired designs can be classified under the heading ‘conceptual’, when the result of the inspiration is an artifact, or ‘computational’, when the result is a process. Both areas face the challenge of identification of relevant biological phenomena, the abstraction of concepts to a level that can be understood by engineers without a background in biology, enabling non-obvious applications of the phenomena, and avoiding misinterpretations of the underlying biological phenomena [48, 49]. Such approaches already widely known and applied like bionics and evolutionary computation. They may become even more important for the product design process, but are not dominant yet.
3. **Ontologies and semantic interoperability**. Ontologies are required for both encoding design knowledge and for facilitating semantic interoperability. Development of engineering ontologies on a large scale can evolve in a similar manner to the compilation of the Oxford Dictionary (see Chap. 10). Researchers (across the globe) could undertake ontology development in selected areas and then contribute to a global repository [50]. This would require the establishment of appropriate standards for encoding ontologies. Here occurs another collision of re-use of knowledge and IP protection, which is still to be resolved.
4. **Natural user interfaces**. Reality-based systems facilitate intuitive human–computer interaction with little user training or instruction. This is evident in the recent upsurge in touch-based personal computing devices like smartphones and

tablet computers, and in gesture-based controls in gaming. The portable and ubiquitous nature of tablet computers make them ideal for collaborative design processes like the recording and progressive documentation of design discussions. It is thus likely that NUIs may prove an important factor towards mass collaboration and the democratizing of the design process. As discussed in Sect. 21.4.4 utilization of a user-friendly common client architecture based on backend services helps reduce training and support effort, in particular in case of a change. Definition of different roles in a sole architecture will foster agility.

Another issue with respect to knowledge and information concerns human involvement. Humans need to be ‘in the loop’, especially in the earlier phases of design. In Chap. 15, it has been established that deterministic thinking is not suitable anymore for complex problems. Emergent behavior cannot be explained sufficiently, because interaction between components and their behaviors is not well understood. As highlighted before, socio-technical modelling approaches are necessary to model and evaluate this emergent behavior.

A final trend in research and practice related to information and knowledge in CE concerns IP and its protection. Increasing cooperation between stakeholders necessitates IP protection and enterprise rights management. As virtual product design increases (see previous section), the risks and consequences associated with IP theft rise commensurately. Methods for patent infringement tracking as well as for IP protection in information and data flow must be developed to a further extent.

28.4.5 Organisation/Structure

Trends in the organization and structure of the CE process involve systems thinking and trans-disciplinarity.

With respect to systems thinking, many modern products have achieved such a level of complexity that their behavior cannot be predicted in a sufficient way, followed by unexpected failures. Typical evidence is given by many recalls done in periodical sequence by automotive OEMs to check, repair or replace critical components or modules which have been identified as potential causes for serious, fatal failures. Such recalls also belong to safety-relevant modules like brakes, steering or lighting and discover the lack of system thinking during the development of complex products. This is dangerous if the process of development of a system splits into an arbitrary number of subsystems without any supportive planning of subsystem partitioning. As the consumer buys a complete product, not individually developed components, a systematic development approach across all disciplines accompanied by system thinking of involved stakeholders is necessary. However obviously necessary this may seem, what we see, at least in practice, is more likely to be a traditional bottom-up approach to development driven by past experience with damage, rather than a preventative top-down approach. Traditional approaches are based either on object-oriented concepts or a function-oriented approach for many reasons [51].

System thinking is a holistic analysis approach that focuses on the way that constituent parts of a system interrelate and how systems work over time and within the context of larger systems [52, 53]. The systems thinking approach contrasts with traditional analysis, which studies systems by breaking them down into their separate elements. Systems thinking can be used in any area of research and science (medical, environmental, economic, etc.). When applied in the engineering context, systems thinking is known as SE. As described in Chaps. 9, 19 and 27, SE still has a large potential, but still difficult to fully employ [54].

The following key trends drive process changes for system thinking [55]:

1. Increasingly complex, global systems of systems: Products become more complex and connected.
2. Emergent requirements: The most appropriate user interfaces and collaboration modes for a complex human-intensive system are not specifiable in advance, but emerge with usage. Forcing users to specify them precisely in advance of development generally leads to poor business or mission performance and expensive late rework and delays.
3. Rapid change: Trying to stay competitive in a world of increasingly rapid changes requires new levels of agility [56], and shorter times between new releases of products and services.
4. High assurance of qualities: At the same time that SE and development need to become more agile, the growing interdependence of systems and people requires systems to have higher assurance levels. It is even harder to get agreement among multiple system owners with widely disparate quality priorities.

The product lifecycle of modern products is defined and influenced by subject matter experts of many disciplines. As explained in Chaps. 4 and 8, it is also the subject of socio-technical considerations. The self-evolving discipline of integration gains a new level of importance as trans-disciplinarity. Trans-disciplinarity covers the deep integration of various forms of research or expertise to create a holistic approach [8]. As the prefix “trans” indicates, trans-disciplinarity concerns issues that are between the disciplines, across the different disciplines, and beyond each individual discipline simultaneously. When for instance contrary targets exist, trans-disciplinarity can help determine the most relevant problems and solution approaches. It shares capabilities like interdependence with the network-centric world [2].

Trans-disciplinarity arises when participating experts interact in an open discussion and dialogue to achieve a trade-off between them. This is difficult because of the overwhelming amount of information involved, incommensurability of specialized languages in each field of expertise and different solution approaches. To progress under these conditions, subject matter experts need not only in-depth knowledge and know-how of the disciplines involved, but skills in moderation, mediation, association, and transfer. Trans-disciplinarity is also a rising subject of education and training, and yields a new occupational profile.

Trans-disciplinarity needs support by appropriate organization, suitable collaboration methods, and powerful IT tools. Collaboration between stakeholders is deemed essential—not merely at an academic or disciplinary collaboration level, but through systematic, repeatable collaboration with people affected by the stakeholders. In such a way, trans-disciplinary collaboration becomes uniquely capable of engaging with different ways of knowing the world, generating new knowledge, and helping stakeholders understand and incorporate the results in a common, integral product.

28.5 Expectations for the Future

In Sect. 28.4 trends in research and practice have been discussed based on the chapters in the parts II and III. Apart of these considerations, current trends in economy and society can and likely will exert a sizeable influence on CE. These trends are mostly either accompanied by or related to information and communication technology (ICT). To stay in pace, CE must be well synchronized with the development of ICT. Below, some recent developments in ICT are briefly discussed, which are expected to influence future CE solutions.

28.5.1 Mass Collaboration

The rapid advancement of Internet of Things (IoT) is pushing the envelope of how CE is being practiced. IoT provides pervasive real-time, and location sensitive data capabilities, forcing engineered products to evolve with new communication and computational features before, during and after they are deployed to markets. One condition for further progress is mass collaboration. Large numbers of people work independently on a single project, often modular in nature, using social software and computer-supported collaboration tools. This idea has been implemented as crowdsourcing, which typically involves an online system of accounts for coordinating buyers and sellers of labor.

On the other hand, customers are regarded as an important information source for product innovation. It is often accepted that more contributions made by customers/clients may bring better innovation results. Recent results have shown that the most significant positive influence on product innovation results is made by external resources such as consultants, commercial labs, or private R&D institutions, rather than customers, clients, or end users [57]. Moreover, the curve-fitting model reveals that different amounts of input information provided by customers/clients/end-users have curved impacts on innovation results. It is suggested that enterprises should properly allocate their research focus on customers and other important factors and carefully handle the information provided by customers.

Nowadays, crowdsourcing is applied widely in various industries. In product design and development, crowdsourcing has been recognized as an effective way to access external resources and to aggregate a crowd's wisdom in order to create more chances to achieve better design concepts [23]. Taking Proctor and Gamble as an example, the most challenging problems are solved by 'InnoCentive', and the problem-solving rate has increased to 30 %. In another example, Dell has set up an idea storm platform to collect comments and suggestions for all Dell products from Internet users. In addition, Wikipedia, Amazon's Mechanical Turk and iStock-Photo.com are all good examples that take advantage of the tremendous numbers of Web users. Therefore, crowdsourcing appears to be a promising way to solicit external resources to improve product competitiveness.

As an effective way to aggregate a crowd's wisdom for product design and development, crowdsourcing shows huge potential for creativity and has been regarded as one important approach to acquire innovative concepts. However, it is still a challenge to make use of crowdsourcing in product design: how can the large number of crowdsourcing concepts be reviewed efficiently? The challenge lies in approaches and methods to improve the efficiency of evaluating crowdsourcing results and to assist designers in identifying promising design candidates for further evaluation. The workload to review crowdsourcing responses manually is very heavy. Moreover, the reliability of evaluation results heavily relies on designers' personal knowledge and experience. Concept screening methods are needed to assist designers in identifying useful responses from crowdsourcing results.

28.5.2 Cyber Physical Systems

The Cyber Physical System (CPS) encompasses the integration of computation and physical processes. CPS comprises embedded computing devices and networks that monitor and control physical processes, with feedback loops when physical processes affect computations and vice versa. Interaction with the physical environment will provide added value with new capabilities and characteristics to systems, while inclusion of physical processes not only increases the complexity of the system but also increases the uncertainties in the behavior of the system [58].

The computation behavior of the system is not only dependent on its internal configuration but is guided by parameters from the physical world. These external factors may either reflect an adaptive mode in which changes in the environment determine changes in the computational system or may be determined by internal imperatives in which a monitoring process examines the computational behavior of a system and determines appropriate evolutionary actions when the system requires modification. This paves the path for the self-evolutionary cyber physical system (eCPS), which automatically detects the changes in external circumstances or internal conditions and is prepared to handle the changes.

A CPS can build networks autonomously and decentralized and optimize itself. It can solve problems in interaction with humans. One of the most important applications is the Smart Factory (known in Germany also by the initiative "Industrie 4.0" which is intended to set the stage for the 4th industrial revolution), which organizes itself using a CPS in a decentralized and near real-time way [59]. By using real-time data the real world could be merged with the virtual world. With such a virtual map of reality opportunities for entirely new business models arise.

CPS platforms build the basis for interoperability of different "internets": internet of people with IoT and internet of services. As such, there are three perspectives on an internet: people are connected in social networks, as already known. Furthermore, machines can be connected as smart objects and use service oriented internet applications. They use software that use data from CPS platforms and human intelligence for achieving solutions in various areas in near real-time in a decentralized way.

Holistic de-centrality is the main challenge for cyber-physical production systems (CPPS) in which organization, services, objects and software are organized in a fully decentralized way. The industry requires such systems for the production of highly customized products in small quantities with high resource productivity and corresponding speed.

The top level of interoperability is considered with the systems of systems in which multiple CPS can combine their autonomous singular capabilities with their own intelligence. Thus, they can evolve entirely new capabilities and develop new services. This level of interoperability remains a vision for facilitating decentralized, autonomous systems development and design with the capability for self-configuration and plug-and-produce.

28.5.3 Big Data

The amount of data around us is growing exponentially and the leaps in storage and computational power within the last decade underpin the unique selling proposition of Big Data of being able to provide better insights into various business processes or everyday life in a novel way. By analyzing large data sets and discovering relationships across structured and unstructured datasets, it is the driving force behind business analytics and marketed as a significant opportunity to boost innovation, production, and competition. Thus, Big Data is a booming topic in the scientific community as well as in the enterprise world [60].

The increasing volume, variety, and frequency of data produced by the IoT will yield the explosion of data for the foreseeable future. With estimates ranging from 16 to 50 billion Internet connected devices by 2020, the hardest challenge for large-scale, context-aware applications and smart environments is to tap into disparate and ever growing data streams originating from everyday devices and to extract hidden but relevant and meaningful information and hard-to-detect behavioral patterns out of it. To reap the full benefits, any successful solution to build context-aware data-intensive applications and services must to be able to make this valuable or important information transparent and available at a much higher frequency to substantially improve the decision making and prediction capabilities of the applications and services [60].

Proper handling of Big Data extracts knowledge acquired through the three past paradigms for scientific research (theory, experiments, and simulations) with vast amounts of multidisciplinary data. The technical and scientific issues related to this context have been designated as "Big Data" challenges and have been identified as highly strategic by major research institutions and large enterprises.

Many of the Big Data challenges are generated by future applications with which users and machines will need to collaborate in intelligent ways together. In the near future information will be available all around us, and will be served in the most convenient way. Technology becomes more and more part of our daily life. New technologies have finally reached a stage of development in which they can significantly impact our lives.

Within the context of CE, a huge challenge concerns the issue of knowledge discovery in databases (KDD), a nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [61, 62]. Intelligent utilization of existing data (e.g., digital manufacturing) provides a new support function for modern product creation processes. Based on planning data, compiled during preceding product emergence processes, products can be evaluated more easily, which leads to a faster and easier attainment of planning and design levels. The feasibility to segment product data in valid subject-specific groups and to map adequate product-specific assembly operations will remain a subject of research.

Big Data will likely bring disruptive changes to organizations and vendors. On a cautionary note, the analysis of Big Data if improperly used may also pose significant issues, specifically in the following areas: data access and policies, industry structure, and techniques. Because large amounts of unstructured data may require different storage and access mechanisms combined with more sensitive data assembled together, it will be more attractive to potential attackers. Otherwise, derivation of user profiles yields issue in protection of privacy. Citizens and Users may become afraid also of Big Data becoming a Big Brother. Application of Big Data requires the issuing of specific rules and regulations as well as the associated control mechanisms to become useful and fruitful.

28.6 Summary

This final chapter has presented an overview of the main properties, means and trends associated with CE. These three aspects are highly interrelated; research and practical challenges generally touch upon all three.

Of the discussed issues, the trend towards increasing complexity and rising cost of product development are to be counteracted by increasing standardization, integration and interoperability. For these developments, it is critical that the savings incurred outweigh the associated costs; efforts to combat complexity and rising costs must be cost- and time effective.

Especially in cases of very high complexity and dynamics, as is often the case in the early phases of a development trajectory, the benefits of CE are more difficult to reap as has been shown by Valle and Vasquez-Buscello [63]. As indicated in Chap. 8, involvement of people in the earlier phases of development is needed, but requires additional methods for facilitating communication and increasing mutual understanding and commitment [25]. Moreover, vision on the outcomes of a particular development phase should be clear: is it a business case for further processing, is it a rough outline of a new product, is it a variant of an existing product? The intended outcome also determines the degree of complexity and dynamics to be handled [20]. Managing complexity and dynamics in a CE context is a research field in its own right, requiring collaboration between social and technical sciences.

To counter the most pressing challenges of CE, comprehensive layers of CE models are needed, including:

- A generic model incorporating the socio-technical dimensions of CE;
- Adapted models for different application areas;
- Specific models for different phases of the development process for different application areas;
- Specific models for different types of new products and services (fully new, adapted, variants);
- Specific models for different types of supply chain and network collaboration (networked, hierarchical, modular).

To find effective, valuable, and usable methods, tools, and techniques for making CE work, a CE approach is needed in research as well, in which the community that is intended to use the results is taken into account also. Achieving workable CE solutions requires a multidisciplinary approach, involving researchers from different disciplines as well as representatives from practice to validate and test research proposals and solutions.

References

1. Sobolewski M (2014) Unifying front-end and back-end federated services for integrated product development. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 3–16
2. Hsu JC (2014) In a network-centric world. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 17–25
3. Blessing Mavengere N (2013) Information technology role in supply chain's strategic agility. Int J Agile Syst Manag 6(1):7–24
4. Stevenson M (2013) The role of services in flexible supply chains: an exploratory study. Int J Agile Syst Manag 6(4):307–323
5. Alguezaui S, Filieri R (2014) A knowledge-based view of the extending enterprise for enhancing a collaborative innovation advantage. Int J Agile Syst Manag 7(2):116–131

6. Benfenatki H, Kemp G, Ferreira Da Silva C, Benharkat AN, Ghodous P (2014) Service-oriented architecture for cloud application development. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 307–316
7. Rezaei R, Chiew TK, Lee SP, Aliee ZS (2014) Interoperability evaluation models: a systematic review. Comput Ind 65(2014):1–23
8. Hirsch Hadorn G, Hoffmann-Riem H, Biber-Klemm S, Grossenbacher-Mansuy W, Joye D, Pohl C, Wiesmann U, Zemp E (2008) Handbook of transdisciplinary research. Springer Science + Business Media B.V
9. Alexiou K, Johnson J, Zamenopoulos T (2010) Embracing complexity in design. Routledge, Abingdon
10. Koh ECY, Caldwell NHM, Clarkson PJ (2013) A technique to assess the changeability of complex engineering systems. J Eng Des 24(7):477–498
11. ElMaraghy W, ElMaraghy H, Tomiyama T, Monostori L (2012) Complexity in engineering design and manufacturing. CIRP Annals Manuf Technol 61(2012):793–814
12. Jacobs MA (2013) Complexity: toward an empirical measure. Technovation 33(2013):111–118
13. Shafiei-Monfared S, Jenab K (2012) A novel approach for complexity measure analysis in design projects. J Eng Des 23(3):185–194
14. Tamaskar S, Neema K, DeLaurentis D (2014) Framework for measuring complexity of aerospace systems. Res Eng Design 25:125–137
15. Wynn DC, Kreimeyer M, Clarkson PJ, Lindemann U (2012) Dependency modelling in complex system design. J Eng Des 23(10–11):718–721
16. Fang C, Marle F (2013) Dealing with project complexity by matrix-based propagation modelling for project risk analysis. J Eng Des 24(4):239–256
17. Peruzzini M, Germani M (2014) Design for sustainability of product-service systems. Int J Agile Syst Manag 7(3/4):206–219
18. McLay A (2014) Re-reengineering the dream: agility as competitive adaptability. Int J Agile Syst Manag 7(2):101–115
19. Yoshimura M (2012) Framework and methodologies for maximising achievements of product designs by collaborative works. J Eng Des 23(9):674–695
20. Unger D, Eppinger S (2011) Improving product development process design: a method for managing information flows, risks, and iterations. J Eng Des 22(10):689–699
21. Rexfelt O, Almefelt L, Zackrisson D, Hallman T, Malmqvist J, Karlsson M (2011) A proposal for a structured approach for cross-company teamwork: a case study of involving the customer in service innovation. Res Eng Des 22:153–171
22. Yang X, Dong A, Helander M (2012) The analysis of knowledge integration in collaborative engineering teams. J Eng Des 23(2):119–133
23. Chang D, Chen CH (2014) Understanding the influence of customers on product innovation. Int J Agile Syst Manag 7(3/4):348–364
24. Moser BR, Wood RT, Hiekata K (2014) Risk management in the design of engineering as sociotechnical systems. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. In: Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 635–646
25. Van Bossuyt DL, Tumer IY, Wall SD (2013) A case for trading risk in complex conceptual design trade studies. Res Eng Des 24:259–275
26. Grantham Lough K, Stone R, Tumer IY (2009) The risk in early design method. J Eng Des 20 (2):155–173
27. Tavčar J, Duhovnik J (2014) Tools and methods stimulate virtual team co-operation at concurrent engineering. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 457–466
28. Mallek S, Daclin N, Chapurlat V (2012) The application of interoperability requirement specification and verification to collaborative processes in industry. Comput Ind 63 (2012):643–658

29. Gendron E, Pourroy F, Carron T, Marty JC (2012) Towards a structured approach to the definition of indicators for collaborative activities in engineering design. J Eng Des 23 (3):195–216
30. Li Y, Zhao W, Ma Y (2011) A strategy for resolving evolutionary performance coupling at the early stages of complex engineering design. J Eng Des 22(9):603–626
31. Levandowski C, Raudberget D, Johannesson H (2014) Set-based concurrent engineering for early phases in platform development. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 564–576
32. Nelaturi S, Rangarajan A, Fritz C, Kurtoglu T (2014) Automated fixture configuration for rapid manufacturing planning. Comput Aided Des 46(2014):160–169
33. Elgh F (2014) Automated engineer-to-order systems a task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3/4):324–347
34. Xu D, Bil C, Cai G (2014) Framework of concurrent design facility for aerospace engineering education based on cloud computing. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 477–486
35. Zeng Y, Wang L, Deng X, Cao X, Khundker N (2012) Secure collaboration in global design and supply chain environment: problem analysis and literature review. Comput Ind 63 (2012):545–556
36. NN (2010) Code of PLM Openness. ProSTEP iVip Association, Darmstadt. http://www.prostep.org/en/cpo.html. Acessed 15 Mar 2014
37. Bondar S, Ruppert C, Stjepandić J (2014) Ensuring data quality beyond change management in virtual enterprise. Int J Agile Syst Manag 7(3/4):304–323
38. Zorgdrager M, Verhagen WJC, Curran R (2014) An evaluation of forecasting methods for aircraft non-routine maintenance material demand. Int J Agile Syst Manag 7(3/4):383–402
39. Andrè S, Stolt R, Elgh F, Johansson J, Poorkiany M (2014) Managing fluctuating requirements by platforms defined in the interface between technology and product development. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 424–433
40. Fernandes J, Henriques E, Silva A, Moss MA (2014) Requirements change in complex technical systems: an empirical study of root causes. Res Eng Des. doi:10.1007/s00163-014-0183-7
41. Schmidt G, Abut H, Takeda K, Hansen JHL (2014) Smart mobile in-vehicle systems: next generation advancements. Springer Science + Business Media, New York
42. Hülsmann M, Fornahl D (2014) Evolutionary paths towards the mobility patterns of the future. Springer, Berlin
43. Geißel O (2012) AMMU automotive mixed mock-up. Konzeption einer neuen Entwicklungsplattform für die Automobilindustrie. PhD thesis, Universität Stuttgart
44. Musa A, Gunasekaran A, Yusuf Y (2014) Supply chain product visibility: Methods, systems and impacts. Expert Syst Appl 41(2014):176–194
45. Sun J, Hiekata K, Yamato H, Nakagaki N, Sugawara A (2014) Virtualization and automation of curved shell plates manufacturing plan design process for knowledge elicitation. Int J Agile Syst Manag 7(3/4):282–303
46. Greboge T, Rudek M, Jahnen A, Canciglieri jr O (2013) Improved engineering design strategy applied to prosthesis modelling. In: Bil C et al (eds) Proceedings of 20th ISPE international conference on concurrent engineering. IOS Press, Amsterdam, pp 60–71
47. Chandrasegaran SK, Ramani K, Sriram RD, Horvath I, Bernard A, Harik RF et al (2013) The evolution, challenges, and future of knowledge representation in product design systems. Comput Aided Des 45(2):204–228
48. Mak TW, Shu LH (2008) Using descriptions of biological phenomena for idea generation. Res Eng Des 19:21–28

49. Wang W, Duffy A, Boyle I, Whitfield R (2013) Creation dependencies of evolutionary artefact and design process knowledge. J Eng Des 24(9):681–710
50. Rosa F, Rovida E, Viganò R, Razzetti E (2011) Proposal of a technical function grammar oriented to biomimetic. J Eng Des 22(11–12):789–810
51. Maurer M, Winner H (2013) Automotive Systems Engineering. Springer, Berlin
52. Demoly F, Yan XT, Eynard B, Gomes S, Kiritsis D (2012) Integrated product relationships management: a model to enable concurrent product design and assembly sequence planning. J Eng Des 23(7):544–561
53. Mueller D, Ganseforth MM (2012) Analysis and modelling of intertemporal relationships in lifecycle design: a case study for investment goods. Res Eng Des 23:191–202
54. Muller G (2013) Systems engineering research methods. Proc Comp Sci 16(2013):1092–1101
55. Boehm B, Koolmanojwong S, Lane JA, Turner R (2012) Principles for successful systems engineering. Proc Comput Sci 8(2012):297–302
56. Azevedo SG, Govindan K, Carvalho H, Cruz-Machado V (2012) An integrated model to assess the leanness and agility of the automotive industry. Resour Conserv Recycl 66 (2012):85–94
57. Chang D, Chen CH (2014) Exploration of a concept screening method in a crowdsourcing environment. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 861–870
58. Ghimire S, Luis-Ferreira F, Jardim-Goncalves R, Nodehi T (2014) Towards self-evolutionary cyber physical systems. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 547–554
59. Bauernhansl T, ten Hompel M, Vogel-Heuser B (2014) Industrie 4.0 in Produktion, Automatisierung und Logistik—Anwendung, Technologien, Migration. Springer Fachmedien Wiesbaden
60. Bessis N, Dobre C (2014) Big data and internet of things: a roadmap for smart environments. Springer International Publishing, Switzerland
61. Wallis R, Stjepandić J, Rulhoff S, Stromberger F, Deuse J (2014) Intelligent utilization of digital manufacturing data in modern product emergence processes. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc. international conference on concurrent engineering. IOS Press, Amsterdam, pp 261–270
62. Tsui E, Wang WM, Cai L, Cheung CF, Lee WB (2014) Knowledge-based extraction of intellectual capital-related information from unstructured data. Expert Syst Appl 41 (2014):1315–1325
63. Valle S, Vazquez-Bustelo D (2009) Concurrent engineering performance: incremental versus radical innovation. Int J Prod Econ 119(2009):136–148

Author Index

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6

Subject Index

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6