A.K. Kamrani
E. A. Nasr
Editors

Collaborative Engineering

Theory and Practice

Springer

Collaborative Engineering

Ali K. Kamrani • Emad S. Abouel Nasr
Editors

Collaborative Engineering

Theory and Practice

Springer

Editors
Ali K. Kamrani
University of Houston
Industrial Engineering Department
E206 Engineering Building 2
Houston, TX 77204-4008

Emad S. Abouel Nasr
University of Houston
Industrial Engineering Department
E206 Engineering Building 2
Houston, TX 77204-4008

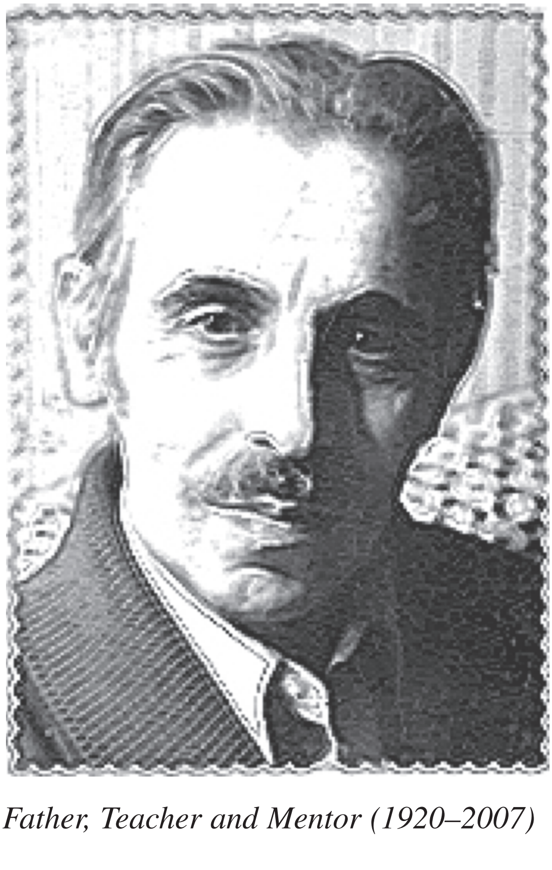
ISBN 978-0-387-47319-2 e-ISBN 978-0-387-47321-5

Library of Congress Control Number: 2008930412

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com



Father, Teacher and Mentor (1920–2007)

Preface

Large or complex design problems often require specialized knowledge from many different fields. Several designers, each with specific core competencies, interact in the design process. During the conceptual phase of the design life cycle, emphasis is mainly focused on establishing the requirements and their proper interaction. The need for integration continues when the design enters the preliminary and detail design phases. Figure P.1 illustrates the scope of the design process for any complex system.

In the virtual, integrated, concurrent design environment, designers interact by sharing information and reaching agreements. By considering proper integration and interaction from the beginning, the problems with the final integration of activities will be significantly reduced.

The surge of the information technology, especially the Internet, provides the infrastructure necessary for an integrated and distributed engineering environment. In this environment, teams of engineers from different parts of organizations could collaborate together toward design, development, and integration. The virtual product design and integration environment would require the collaboration from different teams and functions involved throughout the product life cycle. This environment would require tools that are used for sharing information and knowledge in order to reach an understanding which would contribute to a design which satisfies the customer needs. This environment also provides the means necessary to share tools that are colocated and geographically dispersed.

The distributed environment is highly heterogeneous, where designers, engineers, resources, and models are distributed and not centralized in one location, and groups within the company work together across the computer network. In this environment, many interrelated design decisions are being made in order to meet the objectives. Functions within companies may include design, manufacturing, cost, and life cycle considerations. A collaborative and integrated environment would provide the necessary insight and tools to the designers to quickly construct and reconstruct models of complex products and to evaluate, optimize, and select the better alternative designs. This distributed design environment facilitates the collaborative development of models and exchange of design information and knowledge.

Collaborative engineering (CE) is the systematic approach to the integrated, concurrent design of products and related processes, including manufacturing, product service, and support. This approach is intended to cause developers to consider

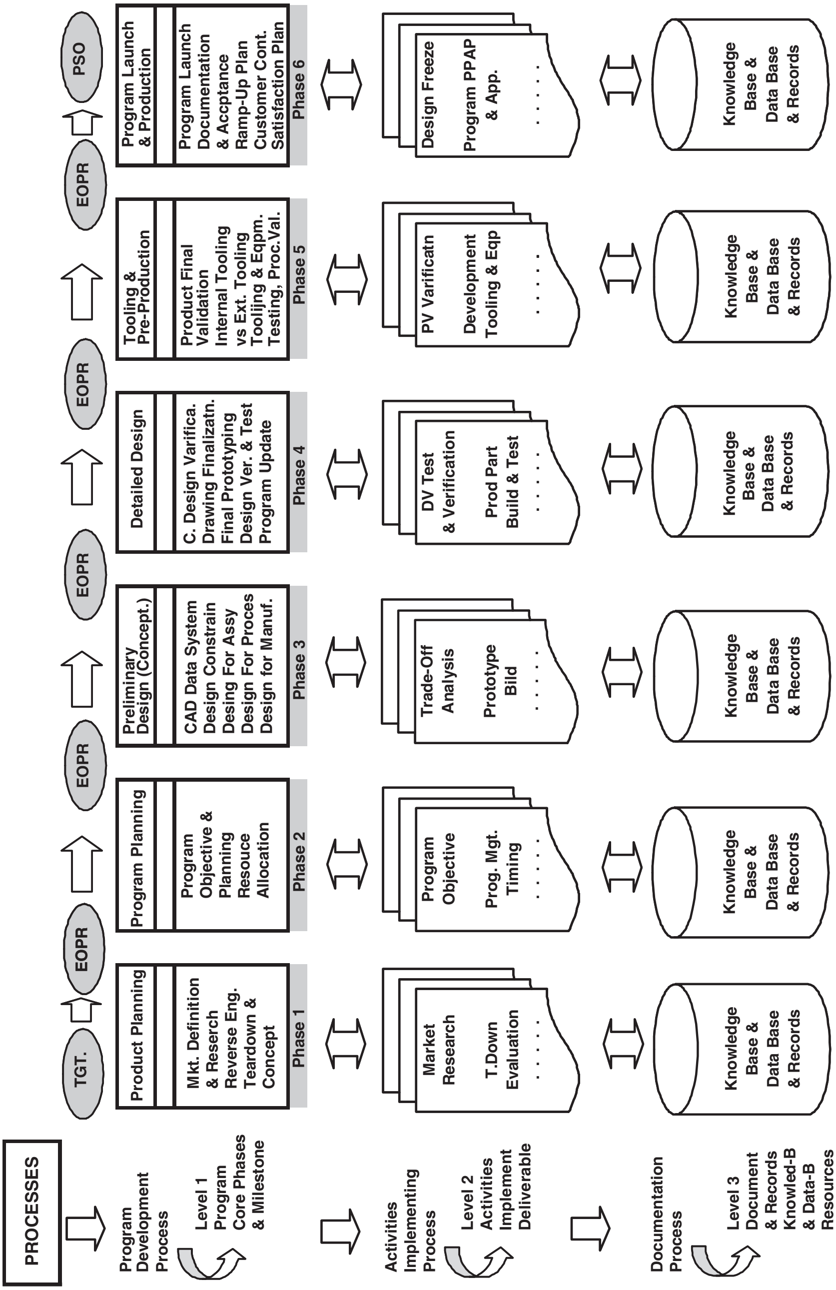


Fig. P.1 Systems engineering design paradigm all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements. The objective of CE is to reduce the development cycle time through a better integration of resources, activities, and processes. This book offers insights into the methods and techniques that enable implementing a CE concept on product design by integrating capabilities for intelligent information support and group decision-making utilizing a common enterprise network model and knowledge interface. The book is a collection of the latest applied methods and technology from selected experts in this area. It is developed to serve as a resource for both researcher and practitioners. It offers the following:

1. Latest research results in CE.
2. Collection of materials from experts in selected topics.
3. Applied methods developed in the field of CE.
4. CE with the emphasis on product design.
5. Discussion on the need and solutions for new engineering paradigm and philosophy required for CE, including the IT infrastructure.
6. Principles and applications of the collaborative environment.

In Chap. 1, Kamrani discusses the requirement for an open and collaborative manufacturing environment. Technologies to support product life cycles and suitable methodologies, tools, and techniques will be the focus of this chapter. In Chap. 2, Kamrani and Vijian provide a discussion on template-based integrated product development life cycle. General guideline for design of a template-based system for integrated design is outlined and a case study for design of electrical motor is provided. In Chap. 3, Feng describes the concept of Six Sigma as a methodology to manage process variations that cause defects, defined as unacceptable deviation from the mean or target; and to systematically work toward managing variation to eliminate those defects. Also, statistical methods for quality improvement, such as statistical models, control charts, process capability, process experimentation, model building, and the evaluation of measurement processes, are discussed. Supply chain workflow modeling using ontologies is presented by Chandra in Chap. 4. One of the primary objectives of supply chain information support system is to develop conceptual design of organizational and process knowledge models which facilitate optimal supply chain management. Data mining (DM) as a process of automatically searching large volumes of data for patterns recognitions is presented by Kamrani and Gonzalez in Chap. 5. DM is a fairly recent and contemporary topic in computer science. However, DM applies many computational techniques which are explained in this chapter. A step-by-step methodology for DM is presented in this chapter. Chapter 6 by Nasr and Kamrani provides an overview of intelligent design and manufacturing (CAD/CAM), the most important reasons of using CAD systems in the manufacturing environment, computer-integrated manufacturing (CIM), the implementation of automation in the production organization, the role of CAD/CAM systems in the manufacturing facility, the CAM cycle in a feature-based design environment, and different types of features. Moreover, this chapter provides a methodology for feature analysis and extraction of prismatic parts for CAM applications is developed and presented. This approach aims to achieve the integration between CAD and CAM. Simulation and optimization are clearly two of the most widely implemented operation research and management science techniques in practice, although several obstacles have limited the acceptance and application of integrated simulation and optimization techniques. In Chap. 7, Asiabanpour, Mokhtar, and Houshman provide a discussion of rapid manufacturing as a technique for manufacturing solid objects by the sequential delivery of energy and/or material to specified points in space to produce that solid. Current practice is to control the manufacturing process by computer using a mathematical model created with the aid of a computer. Also, this chapter discusses the large advantage of rapid manufacturing in speed and cost overhead compared to alternative polymer or metal manufacturing techniques such as powder metallurgy manufacturing or die casting. Moreover, in this chapter, rapid manufacturing as an application of solid freeform fabrication for direct manufacturing of goods is addressed. Unlike methods such as computer numerical control (CNC) milling, these techniques allow the fabricated parts to be of high geometric complexity. In Chap. 8, Assavapokee and Mourtada introduce the basic concept of simulation-based optimization and illustrate its usefulness and applicability for generating the manpower planning of airline's cargo service call center. Because of the continuous increase in oil prices, and combined with many other factors, the airline industry is currently facing new challenges to keep its customers satisfied. In this work, reinforcement learning (RL) and Markov decision process (MDP) are utilized to build and solve the mathematical model to determine the appropriate staffing policy at the airline's cargo service call center.

A robot is a mechanical device that sometimes resembles a human and is capable of performing a variety of often complex human tasks on command or by being programmed in advance. Robotics of the years has seen an immense growth both technologically and otherwise. The recent advances in the manufacturing processes have necessitated the need to enable robots to be more autonomous. Autonomy simply means the ability of the robot to be independent, that is, intelligent. It should be understood that the mimicking of human intelligence and neural function is a relatively nascent research area and has significant strides to overcome in order to achieve this. Chapter 9 by Ibekwe and Kamrani discusses topics related to the design of an autonomous robot. Comprehensive discussion of the modular design, including types of modularity, the characteristics of modular systems, and the development of modular and reconfigurable manufacturing systems, is provided in Chap. 10. In this chapter, Salhieh and Kamrani present a new network-based solution methodology for solving the problem of modularity and classification. Complexity within the manufacturing system comes from the variability and uncertainty in the manufacturing system and from the dynamic nature of the manufacturing environment which increases the number of decisions that need to be made with the difficulty to predict the future response (outcomes) of these decisions. The uncertainty reason can be represented by market demand, product life cycle on the macro scale to tool wear, machine/component breakdown on the micro scale. Variability covers the stochastic nature of manufacturing such as operator performance, work material properties, process repeatability, and supply reliability. The basic elements of the complexity are the absolute quantity of information, the diversity of the information, and the information content. The ability of a production line to produce a product mix requires shop floor monitoring, and information flow giving the required position of the product and the parts for each stage in the production line. A topic related to complexity within the manufacturing system is presented in Chap. 11 by Kamrani and Adat. A simulation-based methodology is presented in order to mitigate the risks associated with manufacturing complexity. Agile manufacturing systems (AMS) will be considered as the next industrial revolution. They are manufacturing and/or management philosophies that integrate the available technology, people, manufacturing strategies, and management systems. Although agility is the set of capabilities and competences that the manufacturing firms need to thrive and prosper in a continuously changing and unpredictable business environment, measuring the level of agility in these firms is still unexplored according to the capabilities and competences. In Chap. 12, Garbie, Parsaei, and Leep present a new solution methodology for manufacturing cell design and analysis.

I thank all my students and other colleagues who participated in this project. I also thank my student Mr. Henry Ibekwe for assisting me in formatting this book. Without his assistance this project would not have been possible. I also thank Springer Publishing (US) for giving us the opportunity to fulfill this project.

Ali K. Kamrani

Contents

List of Authors

Arun Adat
Department of Industrial Engineering, University of Houston, Houston, Texas 77004, USA

Bahram Asiabanpour
Department of Engineering and Technology, Texas State University, San Marcos, Texas 78666, USA

Tiravat Assavapokee, Ph.D.
Assistant Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77204, USA

Charu Chandra, Ph.D.
Industrial and Manufacturing Systems Engineering Department, University of Michigan, Dearborn, Michigan 48128

Qianmei Feng, Ph.D.
Assistant Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004

Ibrahim H. Garbie, Ph.D.
Department of Mechanical Engineering, University of Helwan, Helwan, Cairo, Egypt

Ricardo Gonzalez
Department of Industrial and Manufacturing Systems Engineering, University of Michigan, Dearborn, Michigan 48128

Mahmoud Houshmand
Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran

Henry I. Ibekwe
Department of Industrial Engineering, University of Houston, Houston, Texas 77004, USA

Ali K. Kamrani, Ph.D.
Associate Professor, Department of Industrial Engineering,
University of Houston, Houston, Texas 77004

Herman R. Leep, Ph.D.
Department of Industrial Engineering, University of Louisville, Louisville, KY, 40292, USA

Alireza Mokhtar
Department of Industrial Engineering, Sharif University of Technology,
Tehran, Iran

Ibrahim Mourtada
Manufacturing Engineer, Cameron Measurement Systems, Houston,
Texas, 77032, USA

Emad S. Abouel Nasr, Ph.D.
Mechanical Engineering Department, Faculty of Engineering at Helwan,
Helwan University, Helwan, Cairo, Egypt

Hamid R. Parsaei, Ph.D.
Department of Industrial Engineering, University of Houston, Houston, TX, 77204, USA

Sa'Ed M. Salhieh, Ph.D.
Assistant Professor, Department of Industrial Engineering, University of Jordan, Amman, 11942, Jordan

Abhay Vijayan
Department of Industrial Engineering, University of Houston, Houston,
Texas 77004

Table of Figures

Chapter 1
Collaborative Design Approach in Product Design and Development

Ali K. Kamrani

Abstract This chapter presents an integrated framework for distributed and collaborative environment, which could assist organizations to achieve integrated design goals. The proposed system emphasizes the integration of the software tools and the resources involved in the design process to collaborate the geographically dispersed design teams and vendors. The advancement in information technology (IT) is the driving force for the development of this environment. Also, the early participation of vendors in the design process is considered critical in order to improve the product quality and reduce the development cycle time.

Advances in IT have enabled designers to more effectively communicate, collaborate, obtain, and exchange a wide range of design resources during development [1]. Many manufacturing companies are publishing their product information on the Internet. The network-oriented design environment is a new design paradigm for product development. An integrative framework that enables designers to rapidly construct performance models of complex problems can provide both design insight and a tool to evaluate, optimize, and select better alternatives. Furthermore, a design problem constructed from modeling components made available over Internet might facilitate the collaborative development of analytical system models in addition to the exchange of design information. A well-defined integrated model will predict the required product properties and evaluate alternative solutions in order to meet the defined design objectives and performances.

Key to the analysis of any problem is the identification of what functions are performed and the relationships between them [18]. A collaborative engineering development process includes a set of activities and functions arranged in a specific order with clearly defined inputs and outputs. Each activity in the process will take a set of inputs and transforms it into an output of some value. The process is considered efficient, when the output of the process satisfies the general customer and product requirements and meets management objectives and cost. New technologies and tools along with advancement in IT are helping these organizations in several ways [2, 17]. However, there is no established generic implementation model for wide range of industries.

Associate Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

Software vendors may provide “custom” software packages for individual firms. Different industries have different product development strategies, which demand a generic framework that will help them collaborate efficiently irrespective of their product, organizational structure, and/or geographical location. Two of the more important elements in this changing environment are increased product sophistication and variation. Minimizing the total costs and being quick to develop and market new products is the key for survival. Product development is a complex process requiring expertise from several fields. This will demand integrating the diverse functional areas of an organization on a common platform [22].

In this chapter, an integrative framework that would enable the design teams rapidly construct performance models of complex design problems is presented. This framework can provide both design insight and a tool to evaluate, optimize, and select better alternatives. Interaction between the elements at every level of design is a critical issue. The framework should not be limited only to internal function integration but it should also consider the external functions such as vendors. The vendors have precise and detailed knowledge for their items. This expertise should be incorporated in the main development system to ensure and optimize the product as a complete system. The templates for different processes and/or procedures should be designed systematically to assist in evaluating and optimizing the design alternatives through proper integration and analysis.

1.1 Integrated Product Development

Integrated product development (IPD) is recognized as a critical part of the development of competitive products in today’s global economy. As a company grows larger and products become more complex, hierarchical organizations are established to master the increasingly large organization size, the technical complexity, and the specialization that evolves to master this complexity. This growth also results in the geographic dispersion of people and functional departments. These factors inhibit many of the informal relationships that previously provided effective communication and coordination between functions. A hierarchical organization structure with enterprise activities directed by functional managers becomes incapable of coordinating the many cross-functional activities required to support product development as the enterprise moves toward parallel design of product and process and a focus on time-to-market. Product development teams (PDTs) are a way to address this complexity by organizing the necessary skills and resources on a team basis to support product and process development in a highly interactive, parallel collaborative manner. Some of the basic principles and guidelines for an IPD are as follows:

1. *Understand Customer Needs and Manage Requirements:* Customer involvement increases the probability of the product meeting those needs and being successful in the market. Once customer requirements are defined, track and tightly manage those requirements and minimize creeping elegance that will stretch out development.

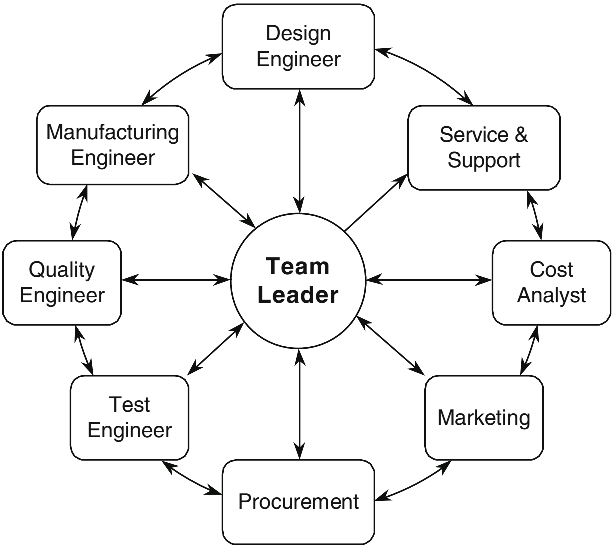


Fig. 1.1 Integrated and collaborative team environment

2. *Plan and Manage Product Development:* Integrate product development with the business strategy and business plans. Determine the impact of time-to-market on product development and consider time and quality as a source of competitive advantage.
3. *Use Product Development Teams:* Early involvement of all the related departmental personnel in product development provides a multifunctional perspective and facilitates the integrated design of product and process.
4. *Involve Suppliers and Subcontractors Early:* Suppliers know their product technology, product application, and process constraints best. Utilize this expertise during product development and optimize product designs.
5. *Integrate CAD/CAM and CAE tools:* Integrated CAD/CAM/CAE tools working with a common digital product model facilitate capture, analysis, and refinement of product and process design data in a more timely manner. Feature-based solids modeling, parametric modeling, and electronic design frameworks facilitate the downstream interpretation, analysis, and use of this product data.
6. *Simulate Product Performance And Manufacturing Processes Electronically:* Solids modeling with variation analysis and interference checking allow for electronic mock-ups. Analysis and simulation tools such as finite element analysis (FEA), thermal analysis, network computer (NC) verification, and software simulation can be used to develop and refine both product and process design inexpensively.
7. *Improve the Design Process Continuously:* Reengineer the design process and eliminate non–value-added activities. Continued integration of technical tools, design activities, and formal methodologies will improve the design process.

Figure 1.1 illustrates the scope of the integrated design team environment.

1.2 Collaborative Design and Development

Design refers to the activities involved in creating the product structure, deciding on the product's mechanical architecture, selecting materials and processes, and engineering the various components necessary to make the product work [26]. Development refers collectively to the entire process of identifying a market opportunity, functional requirements, and finally testing, modifying, and refining the product until it is ready to manufacture. The development of a product is time-consuming, lengthy, and costly.

In a typical product development process the design occurs largely before final full-scale manufacturing. In most of the cases this design is later altered or refined for the manufacturing difficulties, which leads to increased cost and time. The manufacturing department is responsible for estimating the feasibility, cost of building the prospective new product, and modifications, if necessary. If the decision has been taken to outsource some of the components in the final product, the *vendors* come into direct consideration. The vendors become a part of the design team, as they will be contributing toward the development of the final product. Hence, it is very important to consider the vendors' involvement in the design process beginning from the initial stages of the design and development of the product. Thus, in any product design and development scenario the interaction among marketing, engineering, manufacturing, and, in most cases, the vendors is very important. This requirement is met by the application of collaborative product development (CPD) [6, 13, 16, 29].

In the CPD process, the feasibility [25] for product life cycle is analyzed during the early stages of the design process. The expertise from several fields is considered absolutely essential at every stage of the development process. The expertise is grouped in different teams. Each team is responsible for its contribution throughout the process. In the current market trend such teams are mostly dispersed geographically. The need for integration continues when the design enters the preliminary and detail design phases. In an integrated, collaborative design environment, designers interact by sharing information. By considering proper integration and interaction from the beginning, the problems with the final integration of activities can be significantly reduced. In this context, an integrated system is desired to reduce the overall design cycle time by eliminating the repetitive calculations to obtain optimum results. The important stage of collaboration is breaking the barriers among departments and individuals. This open environment increase groups productivity.

CPD practices are recognized as critical to the development of competitive products in today's dynamic market. A hierarchical distribution of work is essential for large organizations and complex products. This structure also results in the geographic dispersion of people and functional areas. The typical integrated environment is shown in Fig. 1.2. PDTs are formed with personnel from various functional departments to support different stages of the development process including the production and services.

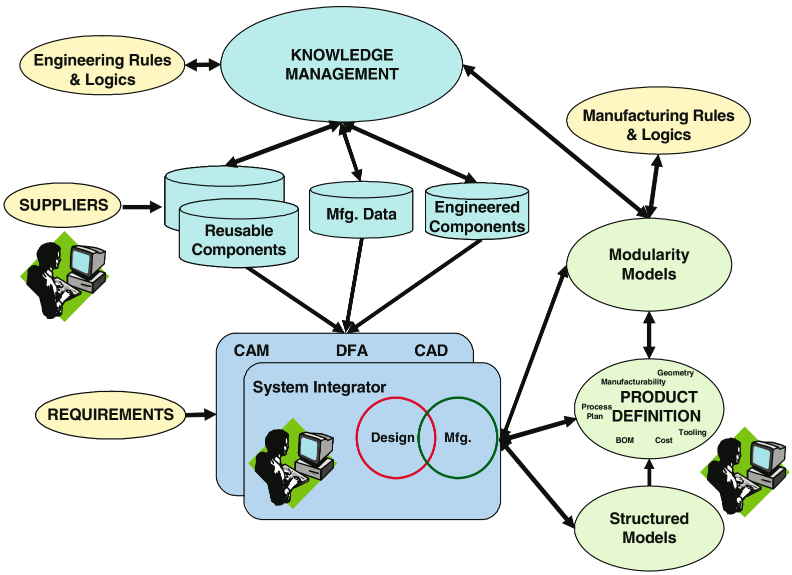


Fig. 1.2 Design team distributed geographically

This early involvement will result in a complete understanding of all the requirements and a consensus approach to the design of both the product and its manufacturing and support processes. PDTs promote open discussion and innovative thinking resulting in superior products, more efficient processes, and, ultimately, a more satisfied customer [12]. The focus of the team will be to satisfy the external customer's product and support requirements as well as the internal customer (functional department) requirements related to factors such as producibility, cost, supportability, and testability.

For an effective distribution of activities among the design teams, the structured approach is very important. The distribution process should start with the product definition and end with manufacturing of the product. The first step toward effective IPD is the understanding and management of customer requirements. The product definition and the structure should be based on the well-understood customer requirements. These requirements should be evaluated and refined considering the expected time-to-market and quality of the product. The early involvement of all the related departmental personnel in the development process provides a multifunctional perspective. This helps reducing and/or eliminating the redundancies of data and manufacturing issues. Today's dynamic market demands more outsourcing from vendors. The utilization of this expertise could help in improving product quality, optimize the product at the system level, and reduce the cycle time [24].

Along with the expertise from different fields and the vendors, the distribution and integration of the resources is also critical. The integration of different CAD/

CAM/CAE tools could shorten the development process and optimizes the design. The use of common electronic product model reduces the chances of redundancy and errors. The feature-based solids modeling, *parametric modeling*, and electronic design frameworks facilitate the analysis and use of product data among participating teams. With the advancements in information technology IT, it is possible to electronically simulate product performance, interface checking, and manufacturing feasibility. This helps in refining the product and the processes from manufacturing perspective [14]. This overall development process should not be set as standard and needs to be improved as it progresses. Figure 1.3 illustrates an example of how design is done using a heterogeneous environment [27, 28].

Central to the design of this system is the interlinking mechanism of the data from different data resources. This link allows for transmitting results from one module to another. The second important part of this design is the parametric databases. Creating the databases is one of the crucial parts in the design of this integrated environment. The databases are built using information (parametric data, models, etc.) supplied by different vendors, design rules, manufacturing, and other pertinent data. Using input module(s) designers participate in the design and analysis process. Designers can input variables and perform necessary calculations and integrated analysis. Figures 1.4 and 1.5 illustrate the scope of integration for the I/O modules and databases.

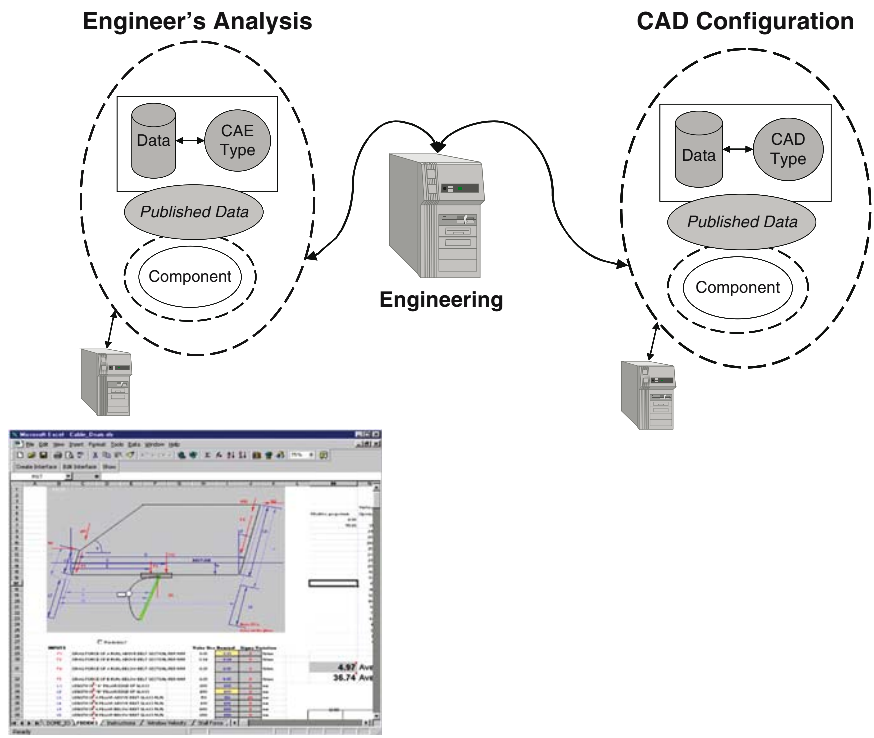


Fig. 1.3 Sample integrated design model

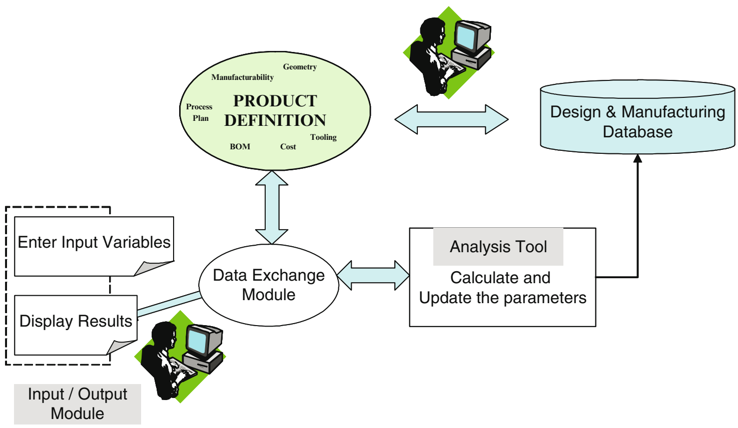


Fig. 1.4 Integration of the I/O modules with the analysis tool

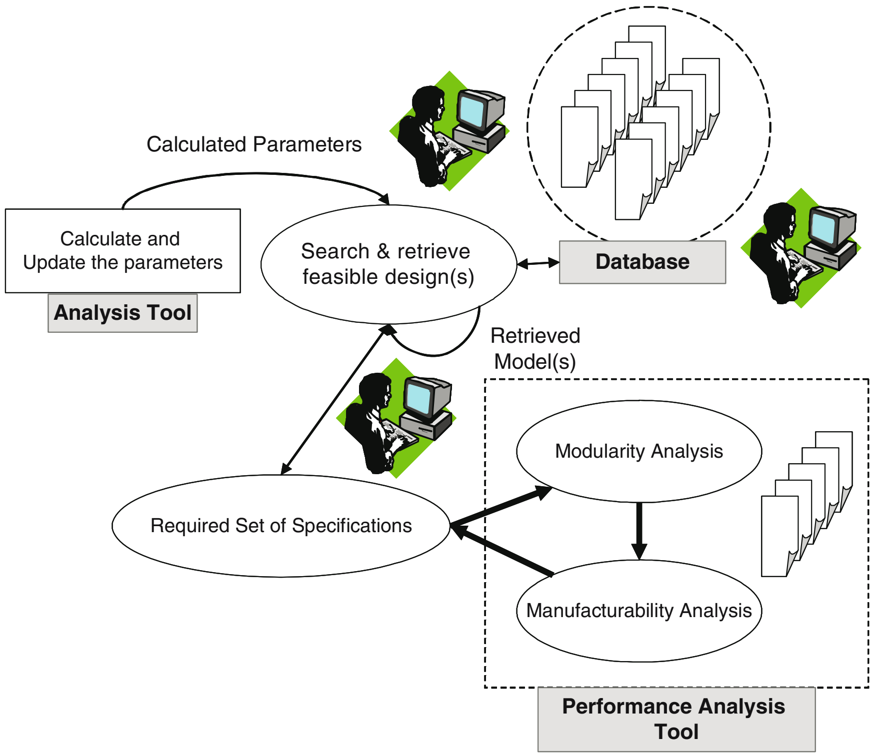


Fig. 1.5 Integration of the analysis tool, databases, and the performance analysis tool

Many researchers are working on developing the technologies or infrastructure to support the distributed design environment. Some are working on providing a platform for sharing or coordinating the product information via the World Wide Web [10, 11].

Others are developing the framework that enables the designers to build integrated models and to collaborate by exchanging services [4, 5, 7, 19–21, 23, 27].

1.3 Case Study: Design of Single-Stage Spur Gearbox

The design problem is decomposed into modules such as physical components, design constraints, parametric models, analysis procedures, and CAD modeling. The important aspect of the framework is an integration of these modules used during the design process in the collaborative environment. The proposed collaborative framework allows the integration, which is capable of revision for the functional model at that instance with any changes made by individuals. The tools used during the product development process vary with the design teams and the vendors. Problems may arise in sharing the information in different forms. Also, sometimes it is very difficult to convert the data or information from one form to the other. This necessitates the *vendors'* involvement at early stages of the development process. The developed system allows the designer to collaborate with the vendors and other team members to optimize the design process considering the relationship within these modules.

1.3.1 System Overview

The problem of gearbox design is decomposed and distributed among the different design teams. At the system level the gearbox design problem has the requirements in terms of design variables and the required output performance characteristics. Along with the design variables there are design constraints like low weight and low cost. Table 1.1 shows the decomposition and distribution of the problem.

The distribution of tasks is done on the basis of the tools used. For this particular design problem different tools and modules considered are analysis tool, optimization team, vendor catalogs, and CAD modeling. These phases represent different design teams that are geographically dispersed. The original equipment manufacturing (OEM) team is the final user of the system. The OEM user enters or changes the design variables and constraints according to the product requirements. The user interface is illustrated in Fig. 1.6.

These variables and constraints are mapped to the *analysis tool* with the use of file wrappers. The *analysis module* is the team of individuals who compose the design problem. The analysis module team performs the numerical analysis and generates the possible alternatives. The design problem and the alternatives are then evaluated at the *optimization module*. The optimization module retrieves the catalogs from the vendor(s) and generates the results in the form of best configuration of the system components.

Table 1.1 Problem decomposition and task distribution

	System level	Subsystem level	Component level
Design variables	Input power Input speed Output power Output speed	Minimum output performance characteristics	
Constraints	Lightweight	Compact design	Gears – Face width – Outer diameter – Shafts diameter
	Low cost	Material Overall size	Material cost Use of standard items
Results	Possible alternatives Overall performance Assembly CAD models	Possible alternatives Trade study	Geometrical detailing Force/stress analysis CAD models

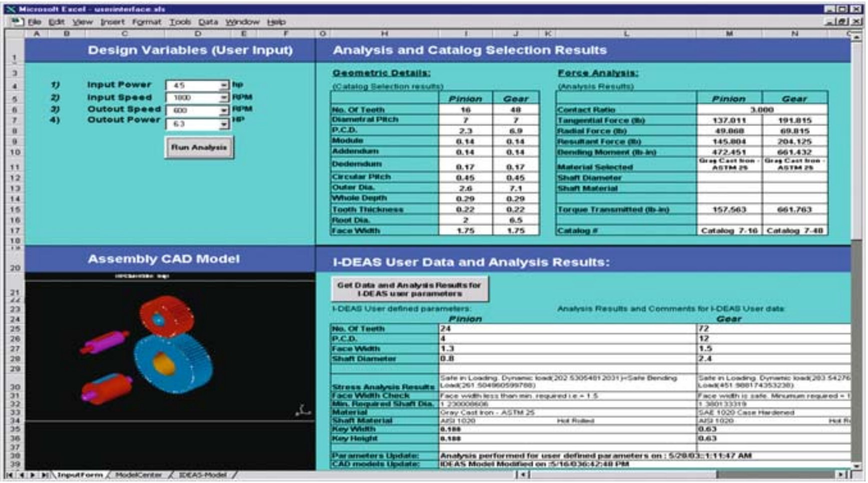


Fig. 1.6 User interface

The *optimization tool* put the catalogs in every instance of the design procedure and evaluates against the design problem and constraint. "Vendor A" is the supplier for the gears and has the set of design catalogs for different gears [*3*, *8*, *9*].

The catalogs are the replaceable modules containing technical specification of the components. The CAD modeling group is responsible for creating the parametric models for the system components. The optimization group sends the results to the CAD station. These results include the geometric details, material specifications, and the analysis data. The CAD modeling group then generates the solid models on the basis of these results. If there are any modifications, the new parameters are sent back to the analysis module for the analysis.

This cycle continues until all the groups approve the design and it meets the functional requirements. Once the design is finalized, the CAD models are created and sent to the user interface.

1.4 System Structure and the Components

The proposed framework is shown in the Fig. 1.7. It consists of five phases: analysis tool, collaborative environment, optimization module, CAD modeling, and vendors' catalogs. An *integrated product* design approach for the design of single-stage speed reducer with a pair of spur gear is structured using the developed system. The design problem is composed of analysis for performance characteristics, catalog selection for vendor supplied items, and CAD modeling for selected alternatives. The product is decomposed into the primary functional system, subsystems, and the components. Maintaining interdependency among these subsystems and components is very important so as to work as a system. It is also important to consider the attributes such as standardization, modularity, future changes, and ease of manufacture and assembly. For the outsource components, the vendors have to contribute with associated design and engineering. In the detail design process, further engineering is done for individual components in the system. The automated generation of

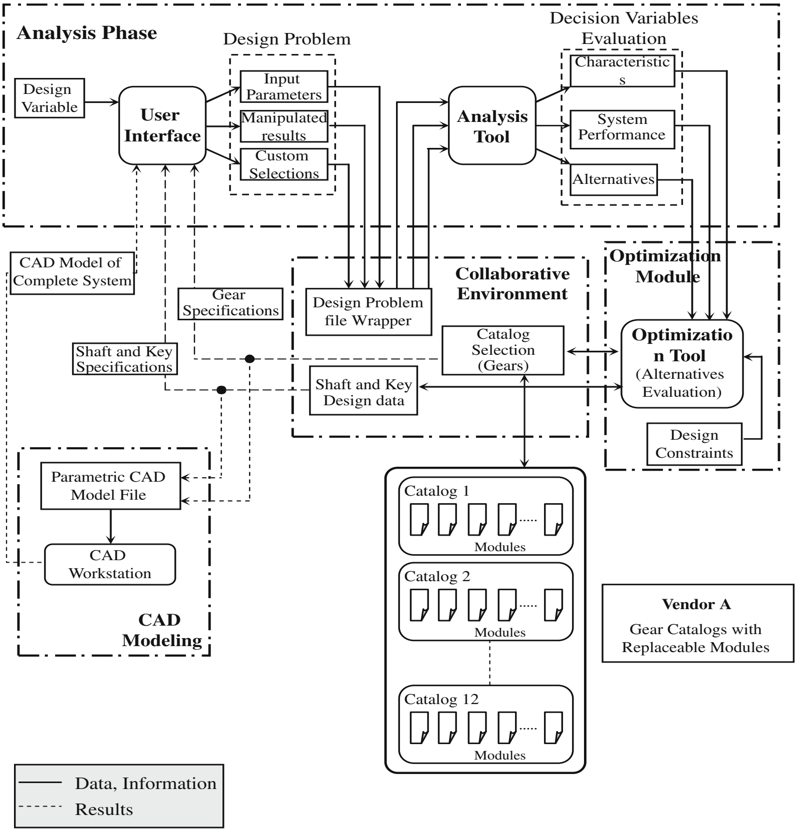


Fig. 1.7 Data and information flow in the proposed integrated system

3-D computer models for different alternatives can serve as unique data model throughout the development process and eliminate and/or reduce the changes of redundancies.

1.4.1 Collaborative Environment

Software integration is a complex process that requires a courtly solution. There has been significant advancement in the data-sharing techniques. The critical issue is maintaining the relationships and dependencies among the different types of data. The data and the information in the proposed system are categorized and defined as modules. The collaborative environment is the integration of all the *modules* and the *catalogs*. The ModelCenter® is used as a service protocol, which connects different modules and catalogs keeping the corresponding relationship as shown in Fig. 1.8.

ModelCenter® is an application that allows users to graphically interact with the modules and generate links between dissimilar applications. During the assembly and the integration of these modules, the design teams can perform the analysis, the optimization, and the trade studies of system level parameters. AnalysisServer® is another application by Phoenix Integration, Inc., that integrates these models by wrapping techniques over the network. The parameters defined in the design problem are the elements of these models.

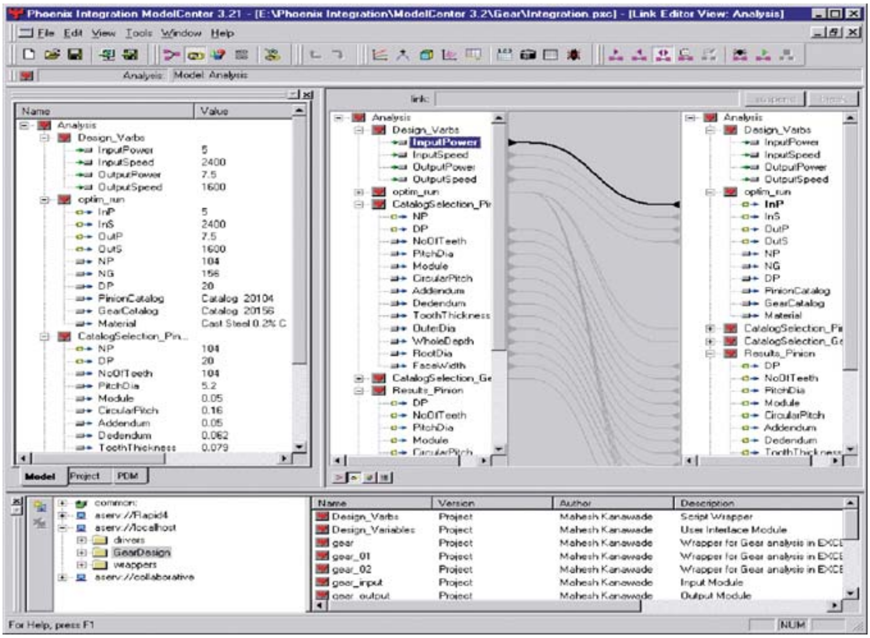


Fig. 1.8 Graphical integration of components

The file wrappers are created to link the input data file (user interface) and the analysis tool. Once the analysis tool calculates the preliminary parameters, the file wrapper maps these values for the optimization module. The results of optimization are returned to the user interface where the user can comprehend these for further changes.

Concurrently, this set of results is sent to the CAD modeling module. Solid models are automatically generated for different components of the product.

1.4.2 Analysis Phase

The design problem is composed of input parameters entered by the user. These are the input and output performance characteristics for the design analysis. The design problem is mapped to variables and synchronized with the analysis tool using the ModelCenter® wrapper file. The analysis tool is introduced to set the decision variables and their evaluation. This application anatomizes the design problem and gives feedback in terms of the performance requirements for different elements in the system. When the user enters/changes the parameters and run the analysis, the wrappers map these as variables and the AnalysisServer® updates the associated variables in the model file. The analysis tool evaluates these parameters and generates the possible alternatives. These alternatives are then further analyzed along with the optimization criterions.

1.4.3 Optimization Phase

The catalogs are the structured databases at the vendors' site. The optimization module team can retrieve these catalogs on limited access basis. Here "limited access" means that the optimization module team can only retrieve the specific information but cannot modify it. During the optimization run the queries are created which retrieve only a particular catalog from the vendor station. It eliminates some of the issues for the information security. The optimization tool maps these replaceable modules from the catalogs for every instance and places them in the current design alternative until the suitable match(s) is found. The optimization module selects the components from the catalogs and returns to the user interface as *results*. This gives the user the detailed specifications for the product and its elements. The results obtained are the parameters obtained from the catalogs, which gives the optimum performance for the given design requirements and constraints. A sample code is listed in Fig. 1.9.

The results obtained from the analysis tool gives theoretical values for the parameters that are acquired from empirical relations. The optimization model is developed to get the configuration, which satisfy the given constraints optimally. The relationships are imposed between the design variables from the design problem and the constraints from the modules. The optimization tool runs an iterative

```
For Each DP In DPrange
 Worksheets(1).Range("C14") = DP
 Calculate
    For Each So In materialRange
     Worksheets(1).Range("C23") = So
     Calculate
      For Each NP In NpRange
      Worksheets(1).Range("C18") = NP
      Calculate
   If b < bU And b > bL And Fb > Fd And Ng < 160 Then
     GoTo EndOptim
     End If
          Next NP
    Next So
Next DP
```

Fig 1.9 Visual Basic optimization code for gear selection

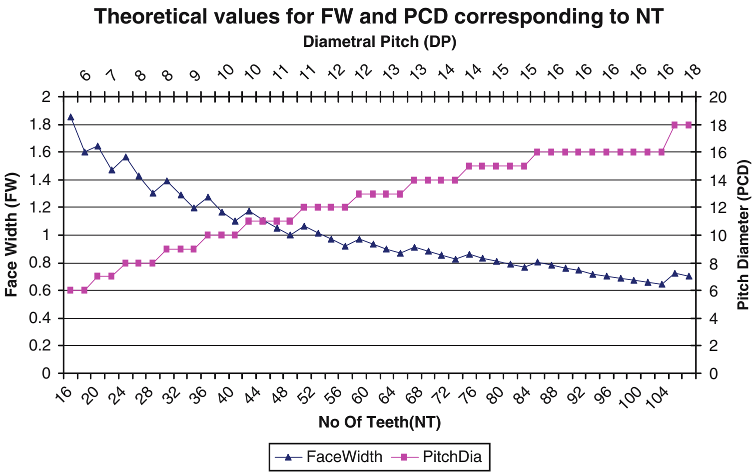


Fig. 1.10 Possible alternatives with expected results

procedure. The modules from the catalogs are retrieved and placed into the current design instance. The modules that satisfy the design constraints and the available components from the vendor catalogs are then selected as alternatives. The graphical comparison of these alternatives is presented at the user interface, which helps the user to trade off for size and cost. Depending on the requirement, the user can then manipulate the design variables to get the best possible configuration. Once the configuration is finalized, the *program files* for CAD modeling are created automatically to generate the solid models of the components. In this case study, a pair of gear is selected from the catalogs and then evaluated for load, power, and other characteristics defined in the design problem. Figure 1.10 illustrates the comparison chart used by the engineer in order to evaluate the alternatives.

1.4.4 Parametric CAD Modeling

Another important module of the proposed framework is *parametric CAD Modeling*. The parametric models for different components are created (Fig. 1.11). The results obtained from the optimization module are used to create the 3-D solid models of each element in the system. These CAD models assist the designer to visualize the interaction of the components for a given configuration. An automated design dramatically reduces the time spent in generating the results for several alternatives.

It also serves as a basis for generating detailed documentation for the manufacturing. A Visual Basic code is used to edit the program files with the newly calculated variables. The parametric models are created for each component in the system for optimized configuration. IDEAS is the CAD tool used in the system. However, any CAD software can be used. For the synchronous, real-time application, parametric models are created [15]. The parametric model allows the user to update the model

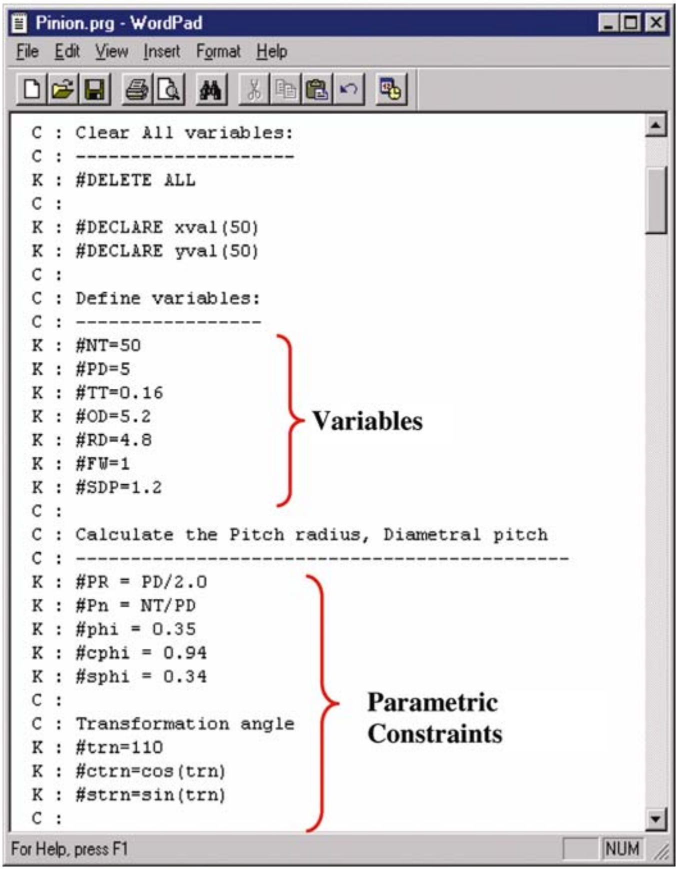


Fig. 1.11 IDEAS macro program file for gear parametric model

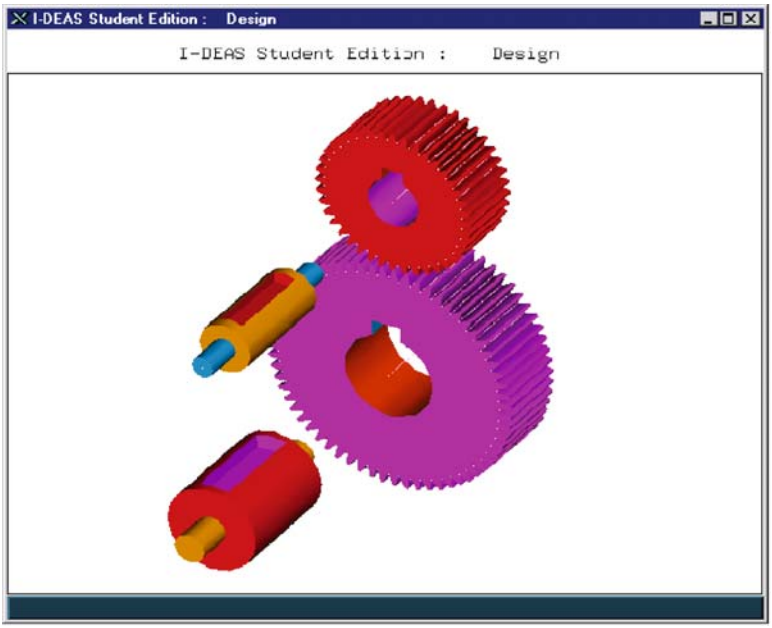


Fig. 1.12 CAD models for the gears and the shafts

according to any change in the design process. For the gear models, pitch circle diameter is a reference dimension and all the other geometric dimensions are constrained and defined using the equations. Once the optimization module generates the alternatives and the results, they are conveyed to the parametric models. The change in pitch circle diameter updates the geometry. The solid models of the components are generated and can be displayed at the user interface in picture format. Solid model for a gear and the respective shaft is shown in Fig. 1.12.

For every change in the design process, the users can retrieve the corresponding CAD models. This gives the designer a chance to visualize the different alternatives and the optimum configuration of the component.

The proposed framework provides the means of integrating software tools that enables the designers to foresee the overall product and enterprise fulfillment during development phases. It will reduce the time required for repetitive analysis for different alternatives. Thus the designer can evaluate more alternatives and can obtain the optimal solution. This integrated system allows the designers to participate in the design process irrespective of geographical location. The developed system provides the capability for design of templates for catalog-based design. Vendors can participate in the development process with their items as catalogs. The optimization phase offers the designers a chance to evaluate different alternatives and the trade-offs.

1.5 Conclusions

Today's manufacturers encounter various difficulties involved in the product development process and these must be overcome for international competitiveness. The obstacles include shortened product life cycle, high-quality product, highly diversified and global markets, and unexpected changes in technologies and customer needs. Any delays in development and you run the risk of losing revenue to your competition. Also the companies are heading toward vendor-based manufacturing (i.e., the manufacturers are trying to get most of the work done by the vendors so as to minimize the time-to-market). Hence it is essential to utilize a computer-aided system in designing, manufacturing, testing, and distributing the products to minimize the time-to-market. For the integration of information at every stage of product development there is need for collaborative technology for job collaboration. As the assistant of the design and development of new products, integrated design technology plays a very important role. The described framework confirms design assumptions and predicts product performance in the early stages of the design process. This results in a faster product development cycle—with lower associated costs—achieved by eliminating the need to constantly build, test, and redesign.

References

1. Albrecht R. and Nicol N. (2002), Microsoft Access Projects with Microsoft SQL server, Microsoft Press.
2. Agnar G., Harry B., and Mariano C. (2004), The implementation process of standardization, Journal of Manufacturing Technology Management, Vol. 15, No. 4, pp. 335–342.
3. Bhandari V. (2000), Design of Machine Elements, Tata-McGraw Hill Publications.
4. Borland N. (1997), DOME—MoDeL Language Reference, MIT, Cambridge, MA.
5. Charles N. (2002), New type of standard for accessibility, designed to foster the competition and innovation of designers, developers, and project and business management, Behavior and Information Technology, Vol. 21, No. 3, pp. 155–169.
6. Chen Y. and Liang M. (2000), Design and implementation of a collaborative engineering information system for allied concurrent engineering, International Journal of Computer Integrated Manufacturing, Vol. 13, pp. 11–30.
7 Cutkosky M., Toye G., Leifer L., Tenenbaum J., and Glicksman J. (1993), SHARE: A Methodology and Environment for Collaborative Product Development, Post-Proceedings of IEEE Infrastructure for Collaborative Enterprise.
8 Deutschman D., Michels J., and Wilson C. (1975), Machine Design: Theory and Practice, Macmillan Publishing Co. Inc.
9. Dudley D. (1962), Gear Handbook: The Design, Manufacture and Applications of Gears, First Edition, McGraw Hill Book Company.
10. Eddy M. and Anthony D. (1999), Web-centric systems: A new paradigm for collaborative engineering, Journal of Management in Engineering, Vol. 15, No. 1, pp. 39–45.
11. Emilda S., Alex L., and Shaik M. (2004), COVES: An e-business case study in the engineering domain, Business Process Management Journal, Vol. 10, No. 1, pp. 115–125.
12. Gupta A. and Wilemon D. (1998), Managing Global Product Development Teams, IEEE—1998, 0-7803-5082-0.

13. Khalid H. (2001), Towards Effective Collaborative Design, Proceedings of HCI International, Mahwah, NJ.
14. Koufteros X., Vonderembse M., and Doll W. (2001), Concurrent engineering and its consequences, Journal of Operations Management, Vol. 19, No. 1, pp. 97–115.
15. Lawry M. (1997), SDRC I-DEAS Master Series Guide, SDRC Press.
16. Lee R., Tasi J., Kao Y., Lin G., and Fan K. (2003), STEP-based product modeling system for remote collaborative reverse engineering, Robotics and Computer-Integrated Manufacturing, Vol. 19, No. 6, pp. 543–553.
17. Manuel C., Pedro C., Carlos V., and Nuria A. (2002), Product data quality and collaborative engineering, IEEE Computer Graphics and Applications, Vol. 22, No. 3, pp. 32–42.
18. Pahl G. and Beitz W. (1996), Engineering Design: A Systematic Approach, Springer Publications.
19. Pahng F., Senin N., and Wallace D. (1998), Web-Based Collaborative Design Modeling and Decision Support, ASME-DETC 1998.
20. Park H. and Cutkosky M. (1999), Framework for modeling dependencies in collaborative engineering processes, Research in Engineering Design, Vol. 11, No. 1, pp. 84–102.
21. Pawar K. and Sharifi S. (1997), Physical or virtual team collocation: Does it matter?, International Journal on Production Economics, Vol. 52, No. 1, pp. 283–290.
22. Rouibah K. and Caskey K. (2003), A workflow system for the management of inter-company collaborative engineering process, Journal of Engineering Design, Vol. 14, No. 3, pp. 273–293.
23. Senin N., Borland N., and Wallave D. (1997), Distributed Modeling of Product Design Problems in a Collaborative Design Environment, CIRP International Design Seminar Proceedings: Multimedia Technologies for Collaborative Design and Manufacturing.
24. Sethi R. and Nicholoson C. (2001), Structural and contextual correlates of charged behavior in product development teams, Journal of Product Innovation Management, Vol. 18, No. 3, pp. 154–168.
25. Srinivasan V., Williams S., Love J., and David B. (1997), Integrated product design for marketability and manufacturability, Journal of Marketing Research, Vol. XXXIV, No. 1, pp. 154–163.
26. Ulrich K. and Eppinger S. (2000), Product Design and Development, Second Edition, McGraw-Hill Publication.
27. Wallace, D. and Senin N. (1997), A Framework for Mixed Parametric and Catalog Based Product Design, MIT, CADLab.
28. Wallace D., Senin N., and Sferro P. (2000), Integrated design in a service marketplace, Computer-Aided Design, Vol. 32, No. 2, pp. 97–107.
29. Wang B. (Ed.) (1998), Concurrent Design of Products, Manufacturing Processes and Systems (Automation and Production Systems: Methodologies and Applications), Vol. 3, Gordon and Breach Science Publishers.

Chapter 2
Template-Based Integrated Design: A Case Study

Ali K. Kamrani[1] and Abhay Vijayan[2]

Abstract This chapter discusses the application of concurrent engineering principles in the product development phase to achieve maximum cost reduction by foreseeing the manufacturability constraints and solving perceived problems beforehand in the concept-generation phase itself. This would entail integrating the different phases, namely, design calculation, computer-aided design, and process planning. This design methodology is applied to the process of shaft design and bearing selection in the product development phase of a motor. The design calculations include a force analysis and also bearing life calculations. The output from these design calculations is used to develop drawings in integrated design and engineering analysis software (I-DEAS). These designs are then used to carry out a time and cost analysis.

2.1 Introduction

The increased globalization of the manufacturing industry opened up new markets with varying customer requirements and also increased competition with more number of companies by introducing low-cost products. This caused an acceleration in the rate of product change which forced engineers and managers to respond with products having lower cost, better quality, and shorter development times. As a result, companies started reinventing their product development methodologies. One of the areas that have gained importance due to this is concurrent engineering. Concurrent engineering techniques have gained wide acceptance in the high volume manufacturing companies such as automotive and consumer electronics markets. It has also gained importance in the low volume, high innovation industry of aerospace. Concurrent engineering is defined as an integrated and systematic approach to the design of products and their related processes, including manufacture and support.

[1] Associate Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004

[2] Department of Industrial Engineering, University of Houston, Houston, Texas 77004

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

This approach causes designers to consider the whole product life cycle from conception through disposal while selecting a design [22]. One of the main reasons for concurrent engineering to gain broad acceptance was its impact on time. Second, it got the manufacturing and marketing departments involved early in the product development process. This enabled them to influence the design and obtain a cost-effective and high performance product. It also made full use of latest advances in information technology to develop libraries with complex accumulations of detailed design. Because of this, knowledge gained during the development of one product could be passed on to subsequent product developments [20].

2.2 Problem Description

Any manufacturing process starts with the product development phase. In a sequential engineering approach the design process and the actual manufacturing are considered as mutually independent. The design department develops a product depending on the customer requirements and design constraints and provides engineering drawings to manufacturing. Here the design department does not take into consideration the manufacturing capabilities and the process limitations on the shop floor. The manufacturing department then carries out a manufacturability analysis of the design and returns it back to the design section with a list of manufacturing concerns. Here each department passes the design or drawings back and forth until they achieve functional success. In the concurrent engineering approach, this back and forth movement of drawings is completely eliminated since manufacturability concerns are addressed in the design phase itself.

Once the design is finalized, resources have to be assigned for manufacturing the product. This is carried out in the process planning phase. In a manual process planning environment, each process plan reflects primarily the experience and skills of the individual process planner; hence it is difficult to maintain consistency and discipline in the planning process [5]. It also acts as a significant break in the automated flow of data from CAD to CAM. Hence it is necessary to automate process planning along with the design phase and allow smooth integration of the two. Advances in information technology and increase in the processing speed of computers help to develop libraries which contain all the data required for manufacturing. This also eliminates the need to develop a new process plan from scratch if a similar product component has already been manufactured.

2.3 Problem Solution

Modifications made to the product in the design phase have maximum impact on the cost and functionality of the product [15]. Hence, we would be concentrating on applying concurrent engineering principles in the product development stage and thus achieve maximum cost reduction by foreseeing the manufacturability constraints and solving perceived problems beforehand in the concept-generation phase itself.

In an integrated CAD/CAM environment, parts are detailed using a computer graphics system. The system stores the geometric information necessary to create process plans. These process plans determine the machining processes, tools, and machines required to convert a part from its initial form to the desired form. To apply this methodology, we have to interlink the different phases of the manufacturing process. The template-based system can be divided into three different parts:

1. Design calculations.
2. Drafting.
3. Process planning.

The system schema is illustrated in Fig. 2.1. The user requirements are analyzed and converted into input parameters for the design calculation phase. The output from the design calculations is utilized as the control parameters for selecting feasible drawings from the CAD system. The process planning module uses the CAD data and then generates route sheets as well as a cost analysis of feasible designs. This enables the designer to study the impact on the design on the basis of cost and manufacturing capability.

The template-based design methodology is used for the bearing selection and shaft design process of an electric motor. The input parameters are speed, power of motor, and also the minimum life of bearing required. This information is then used to obtain the parameters for CAD models. The CAD modeler then returns feasible part drawings which act as the input to the process planning phase. The process planning module prepares the route sheet and carries out a time and cost analysis for different feasible part drawings and returns it to the user.

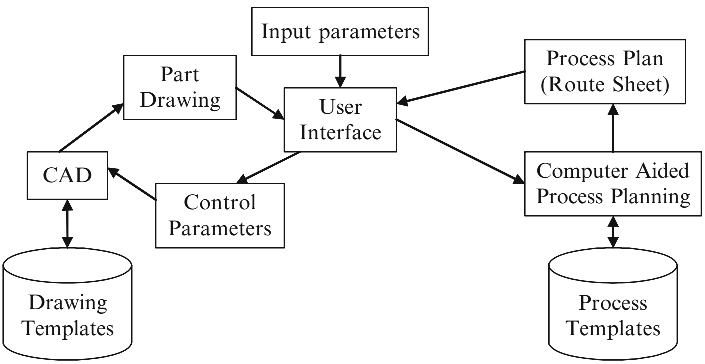


Fig. 2.1 Basic system schema

2.4 Electric Motor

An electric motor converts electrical energy to mechanical energy. Alternating current motors are classified into polyphase synchronous motor, and single-phase or polyphase induction (asynchronous) motors. In polyphase synchronous motors, magnetic field associated with the rotor results from a rotor winding (electric field) excited by direct current via slip rings or from permanent magnets on the rotor structure. In single-phase or polyphase induction motors, the rotor magnetic field is created by electromagnetic induction effects [9]. There are two common features for all types of motors. The first feature is that, for an average torque to be produced, magnetic fields of the stator and the rotor must be stationary with respect to each other. The second feature is, for a specific rotor length, air gap diameter, and speed, there is a maximum average power output rating.

In induction motors, power is transferred from the stationary part (stator) to the rotating part (rotor) [10]. The stator is made of a stack of steel laminations with insulated slots opening to the inside diameter and holding stator coils or windings. The rotor is also made up of slotted steel laminations. The rotor conductors and end rings are formed using a die cast process. The rotor stack length is approximately equal to the length of the stator stack and its outside diameter is smaller than the inside diameter of the stator packet by twice the air gap. This rotor packet is fitted on the shaft. The shaft is supported on the bearings. The end of the shaft where the power is transferred through a pulley or coupler is called the drive end. The other end is called the nondrive end. An external fan is mounted on the nondrive end of the shaft for external cooling.

The proposed design methodology is applied to bearing selection and shaft design of electric motors. The inputs to design calculations are speed, power of the motor, type of drive, and minimum life of bearing required. This information is used to calculate and select the minimum diameter of shaft drive end and also to select the bearings required. Dimensions of the bearings and shaft extension dimensions are then used as the control parameters for CAD template retrieval. The process planning module then carries out a cost analysis for the modified product and returns the minimum cost machine to the user.

2.5 Design Calculations

Power (P) and speed (N) are inputs to the design calculation process. The torque (T) acting on the shaft is given by

$$T_{(\mathrm{Nm})} = [1000 \times 60 \times (P_{(\mathrm{kW})}/N_{(\mathrm{rpm})})]/(2\pi) \tag{2.1}$$

The tangential force (F_t) is calculated as

$$F_{t(\mathrm{N})} = [T_{(\mathrm{Nm})}/(D_{(\mathrm{m})}/2)] \tag{2.2}$$

where D is the nominal diameter of the pulley or the coupler [16].

Based on the types of transmission, the radial force (Fr) and the axial force (Fa) acting on the shaft at the point of mounting is determined [21]. Forces generated at the elements mounted on the shaft produce bending moments and the power transmitted produces shearing stresses. A shaft force analysis is then carried out for the section at which transmission element is mounted. This analysis includes both the maximum shear stress theory and the distortion energy theory [14].

The stresses acting on the surface of a solid shaft are given as follows:

$$\sigma_x = 32M/\pi d^3 + 4F/\pi d^2 \tag{2.3}$$

$$\tau_{xy} = 16T/\pi d^3 \tag{2.4}$$

Assuming that the axial force produces the same kind of stress as the bending moment,

$$\tau_{max} = (2/\pi d^3)[(8M + Fd)^2 + (8T)^2]^{0.5} \tag{2.5}$$

$$\sigma' = (4/\pi d^3)[(8M + Fd)^2 + 48T^2]^{0.5} \tag{2.6}$$

From the maximum shear stress theory, the allowable shear stress is given by

$$\tau_{all} = S_y/2n \tag{2.7}$$

From the distortion energy theory, the normal stress is given by

$$\sigma'_{all} = S_y/n \tag{2.8}$$

where d – diameter of the shaft; F – axial force; M – bending moment; n – factor of safety; Sy – yield strength; T – twisting moment; τ_{max} – maximum shear stress; σ'– normal stress; τ_{all} – allowable shear stress; and σ'_{all} – allowable normal stress.

The factor of safety for maximum shear stress is taken as 1.5 and for maximum normal stress is taken as 2. The material used for the shaft is alloy steel C1045 with density 7830 kg/m^3, whose allowable shear stress is equal to 60 kpsi and allowable normal stress is 92.5 kpsi [7]. Equating τ_{max} with τ_{all} and σ' with σ'_{all}, the minimum required diameter of shaft extension is calculated. The shaft is primarily supported at the bearings. Elements mounted on the shaft include the rotor and the transmission device. A free body analysis of this shaft is carried out and the radial and axial forces acting on the bearings determined. A search heuristic is developed to select all the feasible combinations of bearings from the bearing database that will provide the required life. The process retrieves the bearing dimensions for each bearing and its associated cost. The bearings considered are deep groove ball bearings and roller bearings. The length and diameter of the bearing seat would be the same as the inner diameter and width of bearings used. The power and speed required for the application determine the electrical design of stator and rotor. A database of electrical specifications is created which contains the rotor packet dimensions and its corresponding weight for each combination of power and speed.

2.6 CAD Modeler

Integrated design and engineering analysis software (I-DEAS) is used to generate and store shaft design templates. The shaft consists of the drive end and nondrive end shaft extensions, the drive end and nondrive end bearing seats, and the rotor packet seat. The different dimensions of the shaft are obtained from the design calculations. They include the minimum shaft extension and bearing seat dimensions. The rotor seat dimension is obtained from the electrical design database. The drawings are classified on the basis of the shaft ends as shown in Fig. 2.2. It is either cylindrical or conical. Second classification depends on whether the shaft has extensions on both ends or only one shaft extension. Both end shaft extensions are used either to run a tachometer or to run an external blower at the nondriving end.

Further classifications are based on the applications: low speed high torque and high speed low torque applications. Once the template is selected, necessary changes are made in the dimensions and also additional features like undercuts and threading are incorporated. All the changes made to the template are recorded. These modifications are passed on to the process planning stage.

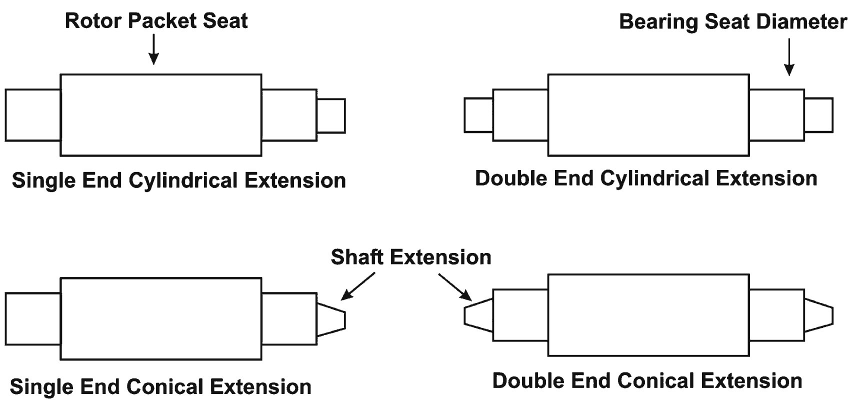


Fig. 2.2 Shaft classifications

2.7 Process Planning

Process planning is the development of a set of work instructions to convert a part from its initial form to the required final form [6, 11, 12, 19]. In the proposed system, the process plan provides detailed description of the manufacturing processes and machine tools required to make the part. This detailed information includes the list of different machines that are used, specific cutting energy for the work material, cost per hour of operation, standard setup and tear down times, and the number of tools utilized for the process. The process plans are saved in the system database. A search heuristic is developed for selecting the process plan. If the process plan does not exist in the database then a new process plan is generated by selecting machines on the basis of the features, dimensions, and tolerance required for the finished product. The process plan information is used to obtain the time required for manufacturing and also for cost estimation.

2.8 Cost and Time Analysis

Each process plan is linked to a cost analysis chart which gives the costs incurred in the manufacturing process. If a new process plan is generated then a detailed analysis for cost estimation is carried out. The time required for manufacturing a part can be classified into productive and nonproductive time [2, 4]. The productive time consists of roughing and finishing operations. Roughing operations are carried out at maximum power and the time taken is given by:

$$t_{mp} = (60 r_v p_s W)/(\rho a W^b) = ((60 r_v p_s)/(\rho a))\ W^{(1-b)} \tag{2.9}$$

where r_v – proportion of initial volume of workpiece to be removed by machining, p_s – specific cutting energy, ρ – density of work material, W – initial weight of work material in pounds, and a, b – constants for machine tool type.

The time taken for the facing operation at maximum power condition (t_{mpf}) is given by

$$t_{mpf} = 60\ V_m p_s / P_m \tag{2.10}$$

where V_m – volume of material removed and P_m – maximum machining power.

The time taken for the finishing operation depends on the surface generation rate R_{sg} for the material. It is the product of speed and feed and is the rate at which machined surface is generated. R_{sg} for alloy steel work piece and high speed steel tool is found to be equal to 23.7 in.^{2}/min [3]. The finished machining time is given by

$$t_{mc} = (60 A_m)/R_{sg} \tag{2.11}$$

where A_m – surface area to be generated.

The tool replacement time is accounted for by modifying the roughing and finishing time equations as

$$t'_{mp} = t_{mp} [1 + (n/(1-n)) (t_{mc}/t_{mp})^{(1/n)}] \quad (2.12)$$

when $t_{mc}/t_{mp} < 1$, and

$$t'_{mc} = ((60A_m)/R_{sg}) (1/1-n)) \quad (2.13)$$

where n is the Taylor tool life index and depends on the material of the tool. Value of n is taken as 0.123 for high speed steel tool [4]. The nonproductive time is calculated as

$$t_{np} = (t_{sa} + n_t t_{sb})/B_s = t_{ln} + n_o t_{pt} \quad (2.14)$$

where t_{sa} – basic setup time, t_{sb} – setup time per tool, B_s – batch size, t_{ln} – loading and unloading time, t_{pt} – tool positioning time, and n_o – number of operations.

The loading and unloading time depends on the weight of the workpiece and is given by

$$t_{ln} = c\, d\, W \quad (2.15)$$

where c and d are constants and have values 38 and 1.1, respectively, for a lathe [8].

The total rate for the machine and the operator is given by

$$R_t = (k_m\, e\, W^f)/(2\, n_y\, n_s) = k_o\, R_o \quad (2.16)$$

where R_o – direct labor rate, k_m – factor for machine overhead, k_o – factor for operator overhead, e, f – constants for a given tool, n_y – amortization period for the cost incurred, and n_s – number of shifts.

The total cost of manufacturing can then be calculated by multiplying the cost rate with the total time required for the process. The manufacturing cost C_{man} is given by

$$C_{man} = R_o (t_{mp} + t'_{mc} + t_{np}) \quad (2.17)$$

when $t_{mc}/t_{mp} \geq 1$, and

$$C_{man} = R_o (t'_{mp} + t'_{mc} + t_{np}$$

when $t_{mc}/t_{mp} < 1$.

Bearings are normally procured from a vendor. If the bearing required can be found in the bearing database then its corresponding cost is retrieved and used in calculating the total cost. If the bearing cost is not available in the database, then the relative cost of the bearing is determined using polynomials [17]:

$$P_3(X) = a_3X^3 + a_2X^2 + a_1X^1 + a_0 \tag{2.19}$$

where a_3, a_2, a_1, a_0, – coefficients, X – standardized dimension, and $P_3(X)$ – third-degree standardized cost function.

The coefficients a_3, a_2, a_1, and a_0 are obtained from the following set of equations.

The total cost for the design including the bearing procurement cost and the manufacturing cost is reported for all the machines that can be utilized to manufacture the shaft. This estimate helps the designer to select the minimum cost alternative along with the machine information on which it can be processed.

$$\begin{pmatrix} \sum_{i=0}^{M}\left(\frac{X_i^{2n}}{Y_i^2}\right) & \sum_{i=0}^{M}\left(\frac{X_i^{2n-1}}{Y_i^2}\right) & \cdots\cdots & \sum_{i=0}^{M}\left(\frac{X_i^{n}}{Y_i^2}\right) \\ \sum_{i=0}^{M}\left(\frac{X_i^{2n-1}}{Y_i^2}\right) & \sum_{i=0}^{M}\left(\frac{X_i^{2n-2}}{Y_i^2}\right) & \cdots\cdots & \sum_{i=0}^{M}\left(\frac{X_i^{n-1}}{Y_i^2}\right) \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{M}\left(\frac{X_i^{n}}{Y_i^2}\right) & \sum_{i=0}^{M}\left(\frac{X_i^{n-1}}{Y_i^2}\right) & \cdots\cdots & \sum_{i=0}^{M}\left(\frac{1}{Y_i^2}\right) \end{pmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{M}\left(\frac{X_i^{n}}{Y_i^2}\right) \\ \sum_{i=0}^{M}\left(\frac{X_i^{n-1}}{Y_i}\right) \\ \vdots \\ \sum_{i=0}^{M}\left(\frac{1}{Y_i}\right) \end{pmatrix} \tag{2.20}$$

2.9 System Components

The structure of the system is divided into four major components: design calculations, bearing selection, design template retrieval, and machine selection with process cost estimation. These components are illustrated in Fig. 2.3. In this system, the product specifications are the input for carrying out design calculations. The external forces acting on the bearings and the shaft extension is calculated. The design calculations also retrieve the electrical design data for rotor packet from the database. A search heuristic is implemented for selecting a minimum cost bearing which can withstand the external forces acting on it. The features required and dimensions of the bearings, rotor packet, and shaft extension are used to retrieve the CAD templates. The template information is used for selecting the machine and carry out a process cost analysis. The output from each component is reported back to the user interface. The system is implemented on a personal computer equipped with a Pentium-4 processor and Windows XP operating system. The graphical user interface, search heuristics, and design calculations are developed in Microsoft Visual Basic (.NET platform). The relational database is developed in Microsoft Access 2003. The 3-D models are generated in I-DEAS and imported into Visual Basic which acts as the drawing templates.

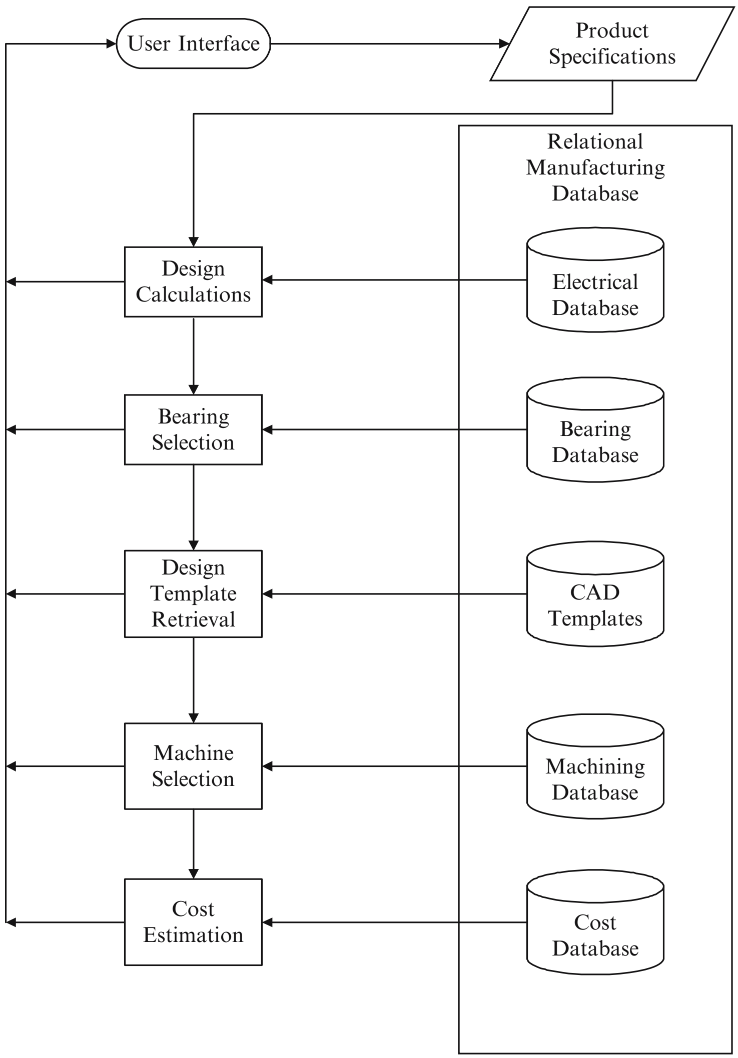


Fig. 2.3 System structure

2.10 Design Calculations

The power and speed entered in the user interface is used to search the electrical database to retrieve specifications of the rotor packet. These specifications are used to calculate the radial and axial magnetic forces acting on the shaft [1]. The radial force acting on the shaft is given by:

$$F_{mr} = (0.1 + 0.005\text{SID})\ [(0.25\text{RL}\cdot\text{SID})/\ (5\text{AG})] \tag{2.21}$$

where F_{mr} – radial magnetic force, SID – inner diameter of stator packet, RL – rotor packet length, and AG – air gap.

The axial force is given by:

$$F_{ma} = (35A.T$$

where F_{ma} – axial magnetic force, SID – inner diameter of stator packet, A – skew angle, T – torque, and AG – air gap.

The skew angle is the angle made by the rotor packet slots with the axis of rotation. Skew angle is provided in some motors to reduce noise levels. Skew angles are generally kept as small as possible since higher skew angles reduce motor performance. A free body analysis of the forces acting on the shaft is carried out to find the external radial and axial forces acting on the bearings as shown in Fig. 2.4. At equilibrium conditions, resultant moment $\Sigma M = 0$, resultant horizontal forces $\Sigma F_{hor} = 0$, and resultant vertical forces $\Sigma F_{vert} = 0$.

So taking moments about the drive end bearing, the radial force acting on the nondrive end is given by:

$$F_{mde} = (-1)[(F_{r2} + G_2 \cos \theta)(l_2 + L) + (F_{mr} + G\cos \theta)li- (F_{rl} + G_1 + \cos \theta)l_1]/L \quad (2.23)$$

The radial force acting on the drive end is calculated as:

$$F_{rde} = F_{r2} + G_2 \cos \theta - F_{rnde} + G\cos \theta + F_{r1} + G_1 \cos \theta \quad (2.24)$$

The resultant axial force is calculated as:

$$F_{ade} + F_{ande} = F_{a1} + F_{a2} + G_1 \sin \theta + G_2 \sin \theta + G\sin \theta \quad (2.25)$$

F_{r1} – radial force acting on the drive end.
F_{a1} – axial force acting on the drive end.

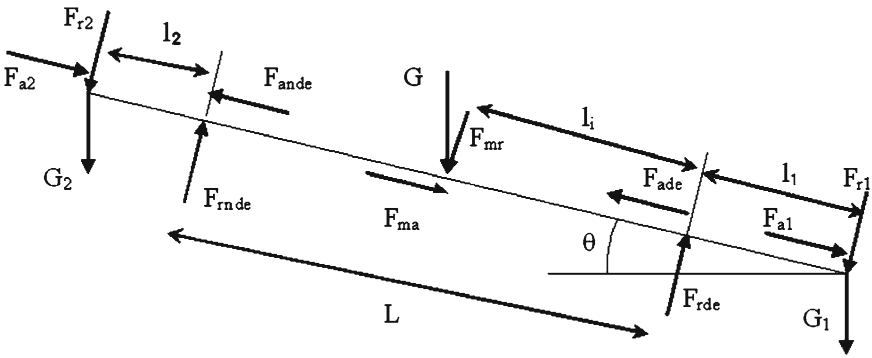


Fig 2.4 Free body diagram

G_1 – weight of drive end transmission device.
F_{r2} – radial force acting on the nondrive end.
F_{a2} – axial force acting on the nondrive end.
G_2 – weight of nondrive end transmission device.
F_{mr} – radial magnetic force.
F_{ma} – axial magnetic force.
F_{rde} – radial force acting on drive end bearing.
F_{ade} – axial force acting on drive end bearing.
F_{rnde} – radial force acting on nondrive end bearing.
F_{ande} – axial force acting on nondrive end bearing.
L – distance between bearings.
l_1 – distance between the drive end transmission device and the drive end bearing.
l_2 – distance between the nondrive end transmission device and the nondrive end bearing.
l_i – distance between the center of the rotor packet and the drive end bearing.
θ – angle made by the shaft with the horizontal plane.

The external forces acting on the bearings and the minimum shaft extension diameter obtained from stress calculations are then called from the design calculations module into the bearing selection module which returns the minimum cost bearings that can be used for the application.

2.11 Bearing Selection

The different types of bearings considered are spherical ball bearings (62 and 63 bearings), cylindrical roller bearings (NU2 and NU3 bearings), and angular contact bearings (72 and 73 bearings) manufactured by Svenska Kullagerfabriken (SKF), Inc. The bearing life required for all applications is 100,000 h. For each of the bearing types, a search is carried out to select the minimum cost bearing. The bearing life calculations for each type of bearing are described in the following sections [18].

2.11.1 Bearing Life Calculations for Spherical Ball Bearings

The equivalent load (P_{eq}) acting on spherical bearings is given by:

$$P_{eq} = XF_r + YF_a \tag{2.26}$$

where F_r – external radial force acting on the bearing and F_a – external axial force acting on the bearing.

$$If[(F_a/F_r) > 0.505\ (F_a/C_o)^{0.231}]\ then\ X = 0.56\ and\ Y = 0.84\ (C_o/F_a)^{0.24} \tag{2.27}$$

or else $X = 1$ *and* $Y = 0$.

where C_o – static capacity of the bearing,

The bearing life (L) in hours is given by:

$$L = 10^6 (C_{dyn}/P_{eq})^3/(60\,N) \quad (2.28)$$

where C_{dyn} – dynamic capacity of the bearing and N – speed in revolutions per minute.

2.11.2 Bearing Life Calculations for Cylindrical Roller Bearings

Roller bearings cannot withstand axial forces but can withstand very high radial forces. The bearing life for cylindrical roller bearings is given by:

$$L = 10^6 (C_{dyn}/P_{Fr})^{10/3}/(60\,N) \quad (2.29)$$

2.11.3 Bearing Life Calculations for Angular Contact Bearings

The equivalent load (P_{eq}) acting on angular contact bearings is given by:

$$P_{eq} = XF_r + YF_a \quad (2.30)$$

where F_r – external radial force acting on the bearing and F_a – external axial force acting on the bearing.

For 72 bearings,

if $[(F_a/F_r) > 1.14]$ *then* $X = 0.35$ *and* $Y = 0.57$ *or else* $X = 1$ *and* $Y = 0$.

For 73 bearings,

if $[(F_a/F_r) > 1.14]$ *then* $X = 0.35$ *and* $Y = 0.93$ *or else* $X = 1$ *and* $Y = 0.55$.

The bearing life (L) is given as

$$\mathrm{L} = 10^6 (\mathrm{C}_{dyn}/\mathrm{P}_{eq})^3/(60\,\mathrm{N}) \quad (2.31)$$

2.12 Bearing Search

The search function retrieves the minimum cost bearing from each type for drive end and nondrive end. The constraints for the search are that selected bearings should be able to provide a minimum of 100,000 h and the inner diameter of the bearing should be greater than the minimum shaft extension diameter required. The minimum cost bearings of each type are displayed in the user interface. Also the combination of bearings for the drive end and the nondrive end which has minimum combined cost is chosen and displayed as the recommended bearings. The following pseudo code describes the search function.

Begin
For *each type of bearing*
Sort bearings in descending order of their cost
Do while *inner diameter of bearing ≥ minimum shaft extension diameter*
If *bearing life calculated ≥ 100,000 hours and*
Cost of bearing ≤ minimum cost bearing then
Select bearing dimensions and
Initialize minimum cost = bearing cost
End *Do Loop*
Select minimum cost combination for drive end and non drive end bearing
End

The dimensions and cost of the bearings are transferred from this module to the design template retrieval module.

2.13 Design Template Retrieval

The 3-D models are generated in I-DEAS and are then imported as image files into Visual Basic. The graphical user interface in Visual Basic prompts the user to specify the type of shaft extension. It also asks if threading is required and also the minimum tolerance to be achieved on the shaft. The features considered in the case study are based on the shaft extension. The different options are single- or double-shaft extensions, conical shaft extensions, and threaded shaft extensions. The system retrieves the dimensions of the rotor packet, which are the internal diameter and the length of the rotor packet, and shaft extension dimensions from the design calculation module. It also retrieves the drive end and nondrive end bearing dimensions, which are the inner diameter and the length of the bearings, from the bearing search module. Drawing templates are retrieved on the basis of the features specified by the user. The selected features and the minimum tolerance to be achieved are then used by the system to choose the appropriate machines. The dimensions of the bearings, shaft extension dimensions, and rotor packet dimensions dictate the final dimensions of the shaft and are passed on to the machine selection and the process cost estimation module. The final dimensions and the features to be processed on the shaft form the basis for the final process cost estimation.

2.14 Machine Selection and Process Cost Estimations

The dimensions and features from the design module are obtained and are used to select a machine and estimate the cost of production. The system searches through the existing process plans in the database to obtain a shaft with matching features and dimensions. If a similar shaft has been previously manufactured, the system returns the machine on which it was manufactured as well as the time and cost required for manufacturing. The pseudo code shown below is used to search through the process plan database.

Process Plan Search
Begin
Do While *processplanreader.Read()*
  If
  New shaft dimensions = processplanreader (dimensions)
  And New shaft features = processplanreader (features)
  And New shaft tolerance = processplanreader (tolerance)
  Then
  machine = processplanreader (Machine)
  cost = processplanreader (Cost)
  End If
End While Loop
End

If a process plan is not found in the database, a detailed analysis is carried out to select the machine and obtain the cost to manufacture. The flowchart for machine selection and cost estimation is given in Fig. 2.5. The system searches for a suitable raw material. The raw material database consists of various sizes of available cylindrical billets. The raw material used in the case study is alloy steel C1045. The machine selection module computes the total length and the maximum diameter of the designed shaft from the dimensions retrieved from the design module. It then searches the raw material database to select the billet, such that minimum material removal will be required for obtaining the final shape. This reduces the processing cost as well as the raw material cost. The following pseudo code describes the raw material search process.

Raw Material Search
Begin
  Sort material data in decreasing order of size
Do While *Rawmaterialreader.Read( )*
  If *rawmaterial length> total length of designed shaft*
    And
      Rawmaterial diameter>maximum diameter of designed shaft
  Then
    select raw material
  End If
  Go to *next smaller raw material*
  cost = processplanreader (Cost)
End While Loop
End

The basic machining processes considered for this system are facing, rough turning, finish turning, taper, and threading. The system selects machines that can

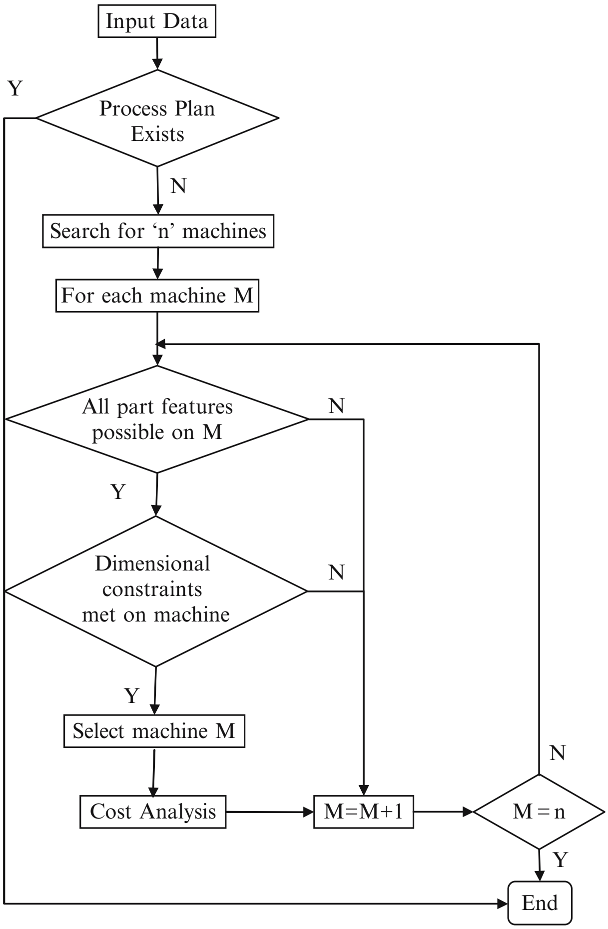


Fig. 2.5 Machine selection module

process the user-defined features on the shaft. The other two constraints for machine selection are the minimum tolerance required and the capacity of the machine. For each machine selected, the time required for machining is calculated. The nonproductive time (t_{np}) required for the process is found by assigning values to the number of operations (n_o) and the number of tools (n_t) required for the machining process which depends on the features to be machined. The setup times for the machine (t_{sa}), the setup time per tool (t_{sb}), and the tool positioning time (t_{pt}) are obtained from the machine database. The batch size is directly obtained from the user interface. The loading and unloading time depends on the weight of the work piece (W), which is obtained from the raw material database. For roughing operations, which are carried out at maximum power conditions, the specific cutting energy (p_s) is found to be 1.3 hp/(in.3/min) for alloy steel work piece and high speed steel tool [3]. The values for tool constants (a, b) are obtained from the machine database.

The time required for finishing operations is found by calculating the surface area generated in the machining surface. The surface area generated depends on the geometry and dimensions obtained from the design template. Tool replacement costs are accounted for by incorporating the Taylor's tool life index in the roughing and finish machining time equations. The factors "k_m," "k_o," "e," and "f" are retrieved from the machine database to calculate the machine rate. The total time required for machining is then multiplied with the machine rate to obtain the cost for machining.

The module then returns the list of machines that can be used with the machining cost for each machine along with the selected machine. Even if the process plan exists in the database, the system runs a detailed cost analysis to verify the cost. This ensures that any changes made to the machine database after the existing process plan was created would be considered for the evaluation. If there is any change in the cost of machining, the modified value can be incorporated in the process plan database.

2.15 Case Studies

In this section, two examples are presented to demonstrate the integrated system and illustrate the operation procedure for the system. The user interface consists of four different windows, each representing a single module. The executable file is coded in Visual Basic and is connected to MS Access databases. The 3-D models are imported as image files and are retrieved by the system [13].

2.15.1 Case Study 2.1

The data required for the design procedure is input in the user interface as shown in Fig. 2.6. In the example, the power required is 100 kW at 1500 rpm. The motor is at an incline of 10 degrees with the horizontal plane. It is a double-ended motor with transmission equipment at both ends. The transmission equipments are spur gear with a nominal diameter of 350 mm and weight 30 kg and direct coupling with a nominal diameter of 300 mm and weight 30 kg. The number of units required is 10, which is considered as the batch size for process cost estimation.

When the "Calculate" button is pressed, the system finds the resultant forces acting on the bearings, which are given below.

Resultant radial force on the drive end = 12,093.16 N
Resultant radial force on the nondrive end = 2,230.44 N
Resultant axial force = 161.87 N

These values are then passed on to the bearing selection module which displays the minimum cost bearings for each type that will provide the required bearing life. This screen is displayed in Fig. 2.7. The bearings recommended by the system are

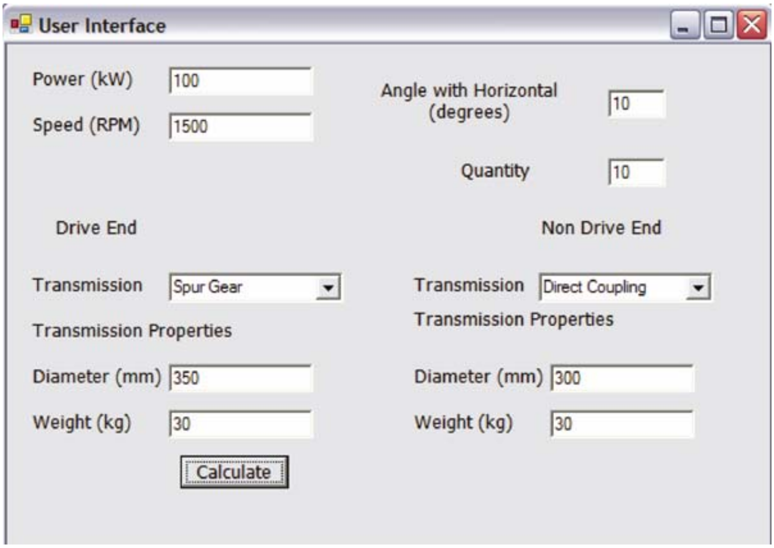


Fig. 2.6 User interface for Case Study 2.1

NU219 with a cost of \$248.98 for the drive end and 6217 with a cost of \$120.59 for the nondrive end. When the "Manufacture" button is clicked, the dimensions of these bearings are passed on to the design module where the user is prompted to specify the features and the minimum tolerance required as shown in Fig. 2.8. In the design template module, the user specifies the shaft end feature as "Double cylindrical shaft end" and clicks the "Template" button to retrieve the template. The user also specifies that no threading is required on the shaft ends. Also the closest tolerance required is specified as 2 μm. This information is transferred to the machine selection module when the "Machine" button is clicked. The machine selection and process cost estimation menu is shown in Fig. 2.9. If a shaft with similar features and dimensions has already been processed, then its corresponding process plan along with the cost is retrieved. The machine selection module displays a list of machines and the corresponding machining costs. The system selects the machine "M3" with the least machining cost "\$ 31.66." It also retrieves the raw material cost from the raw material database and the bearing cost from the bearing search module and calculates the total cost as "\$ 642.79." This cost can then be compared with that of the existing process plan.

2.15.2 Case Study 2.2

In this example, the power required is 220 kW at a speed of 750 rpm. It is a single shaft end motor with a conical shaft end with a flexible coupling as the transmission device. The nominal diameter of the flexible coupling is 400 mm and the weight of

Fig. 2.7 Bearings for Case Study 2.1

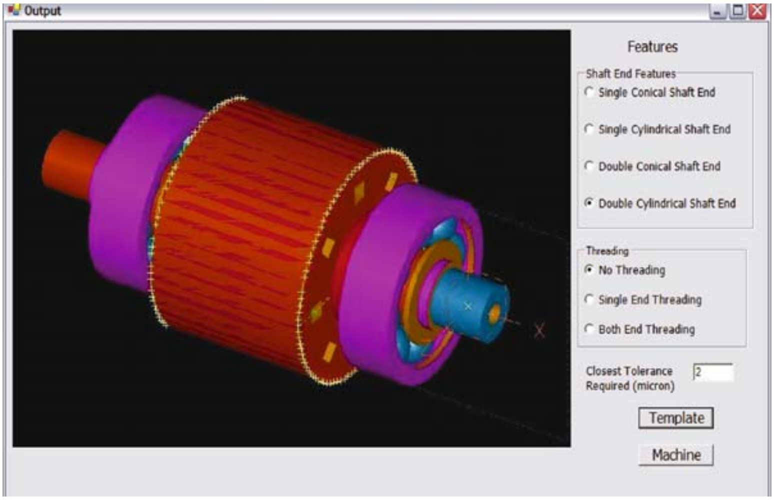


Fig. 2.8 Design template for Case Study 2.1

the coupling is 40 kg. The user interface is shown in Fig. 2.10. The resultant forces acting on the bearings are as given below:

Resultant radial force on the drive end = 4,796.10 N.
Resultant radial force on the nondrive end = 14,649.16 N.
Resultant axial force = 0 N.

Based on these values, the system selects bearings NU219 with a cost of $248.98 and 6219 with a cost of $183.43. This is shown in Fig. 2.11. The bearing dimensions are then transferred to the design template module. The single conical shaft end with the threading option is chosen and the tolerance is input as 3 μm. The design template menu is shown in Fig. 2.12.

Fig. 2.9 Machine selection and cost estimation for Case Study 2.1

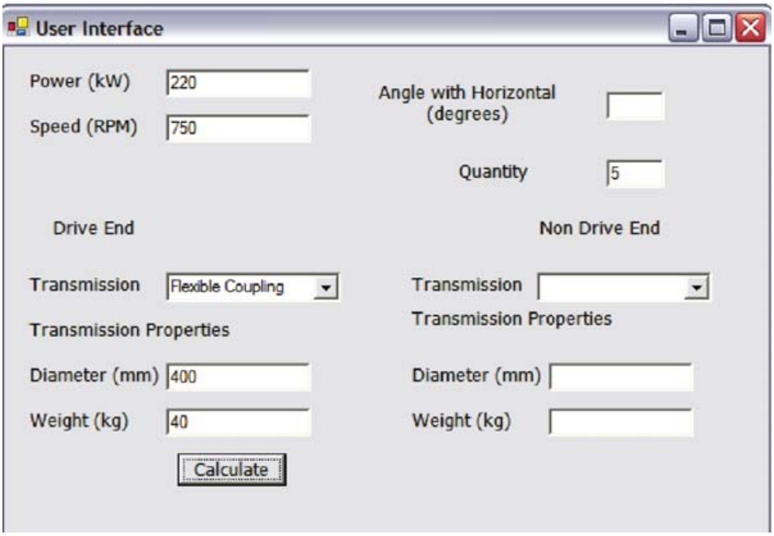


Fig. 2.10 User interface for Case Study 2.2

Based on the input from design template, the system retrieves three machines and calculates the total cost of manufacturing. The machine selection module is shown in Fig. 2.13. In Case Study 2.1, four machines were retrieved instead of three in Case Study 2.2, even though the tolerance was closer in Case Study 2.1. This is because not all machines have threading capabilities. The system was not able to find an existing process plan for the new shaft so the existing plan does not return any value. The machine selected is "M4" with a machining cost of "$41.96." The total cost is found to be "$715.93."

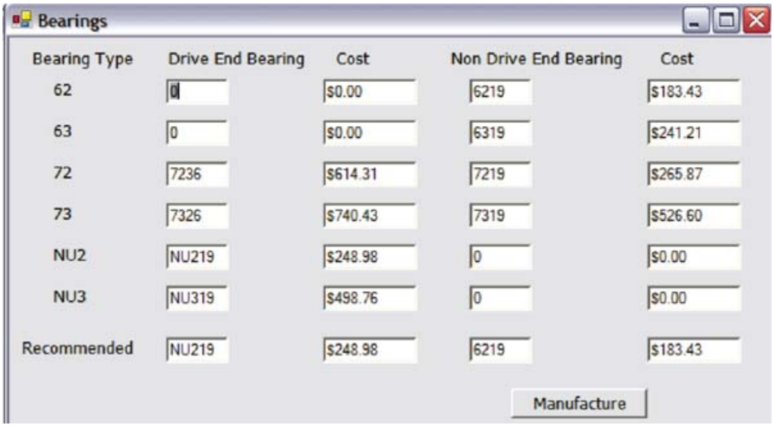


Fig. 2.11 Bearings for Case Study 2.2

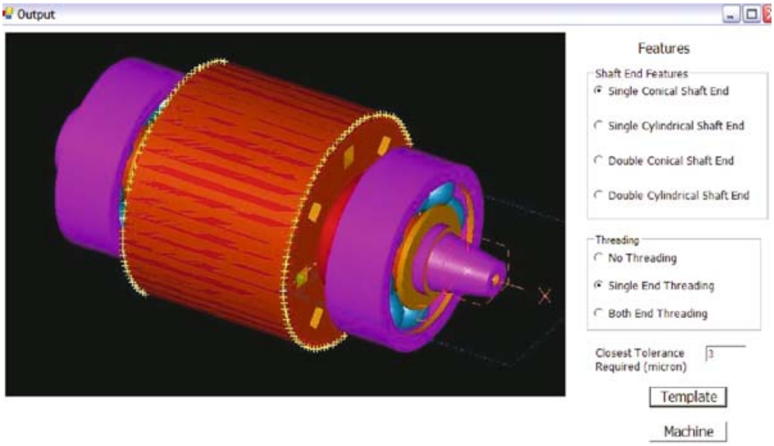


Fig. 2.12 Design template for Case Study 2.2

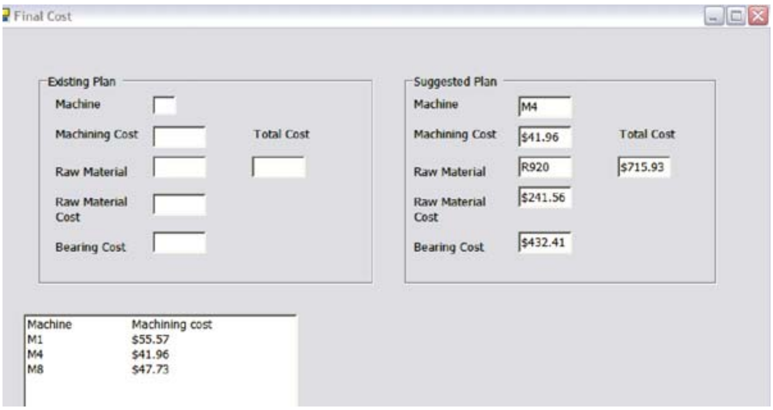


Fig. 2.13 Machine selection and cost estimation for Case Study 2.2

2.16 Conclusion

In this chapter, the implementation of an integrated design based on templates is presented. Concepts used in developing the system are independent of the software used. This system is thus focused on the small- and mid-segment industries which do not have enough resources for costly software upgradations. By using templates, the time required for new product development is drastically reduced. At the same time, incorporating computer-aided process planning into the system gives the designer a better understanding of the cost implications of the modified design with respect to manufacturing. The primary objective was to develop an integrated product and process design through a template-based approach. This system would thus act as an effective tool to reduce cost by foreseeing manufacturability constraints in the concept-generation phase itself. The concurrent engineering approach is applied to product and process development and thus the different phases of manufacturing are integrated together. The system developed has four main components, which are the design calculations, bearing search, and machine selection with process cost estimation. All these four components are connected to the relational manufacturing database. This database contains all the manufacturing information required for product and process development. Each component of the system is developed as a stand-alone module which interacts with the other modules and exchanges information when required. Each module provides the user with an output which acts as the feedback at each stage of product and process development. Since each component is independent of each other, they can be modified without affecting the other components.

The major challenge in implementing this system is that any changes in the manufacturing facility have to be incorporated in the process plans stored. This can be a tedious job but can be overcome by using the hybrid process planning approach instead of the variant-based approach.

References

1. Bartheld, R. G. (2004). Regulation of Noise: Typical Levels. In: H. A. Toliyat and G. B. Kliman (eds.), *Handbook of Electric Motors* (2nd edn., pp. 555–560). New York: Marcel Dekker, Inc.
2. Black, I., Ritchie, J. M., and Walsh, D. S. (1991). Simple turned part design and manufacture with parametric CAD/CAM techniques. Computer-Aided Engineering Journal, 8(4), 147–152.
3. Boothroyd, G., and Dewhurst, P. (1987). *Product Design for Assembly*. Rhode Island: Boothroyd Dewhurst, Inc.
4. Boothroyd, G., and Knight, W. A. (1989). *Fundamentals of Machining and Machine Tools*. New York: Marcel Dekker, Inc.
5. Burgess, J. D. (1984). A review of computer-aided process planning systems. In: J. Tulkoff (ed.), *CAPP: From Design to Production* (1st edn., pp. 3–13). Society of Manufacturing Engineers, Dearbom, Michigan.

6. Chang, T. C., Wysk, R. A., and Wang, H. P. (1991). *Computer-Aided Manufacturing*. Englewood Cliffs, NJ: Prentice Hall.
7. Datsko, J. (1978). *Materials in Design and Manufacture*. Ann Arbor, MI: J. Datsko Consultants.
8. Fridriksson, L. (1979). Non-productive time in conventional metal cutting. Report No. 3. Design for Manufacturability Program. Amherst, MA: University of Massachusetts..
9. Hamilton, H. B. (2004). Types of motors and their characteristics. In: H. A., Toliyat and G. B., Kliman (eds.), *Handbook of Electric Motors* (2nd edn., pp. 26–28). New York: Marcel Dekker, Inc.
10. Hoffmeyer, W. R., Martiny, W. J., and Johnson, J. H. (2004). Induction motors—Polyphase and single phase. In: H. A. Toliyat and G. B. Kliman (eds.), *Handbook of Electric Motors* (2nd edn., pp. 35–43). New York: Marcel Dekker, Inc.
11. Houtzeel, A. (1996). Computer-aided process planning. In: J. M. Walker (ed.), *Handbook of Manufacturing Engineering* (pp. 461–480). New York: Marcel Dekker, Inc.
12. Hundal, M. S. (1993). Rules and models for low-cost design. Design for Manufacturability, 52, 75–84.
13. Kamrani, A., and Vijayan, A. (2006). A methodology for integrated product development using manufacturing templates. Journal of Manufacturing Technology Management, 17(5), 656–672.
14. Natrajan, R. N. (2001). Machine design. In: L. L. Faulkner and E. Logan Jr. (eds.), *Handbook of Machinery Dynamics*. New York: Marcel Dekker, Inc..
15. Nevins, J. L., and Whitney, D. L. (1989). *Concurrent Design of Products and Processes*. New York: McGraw-Hill.
16. Newell, C. J. (2004). Mechanical considerations. In: H. A. Toliyat and G. B. Kliman (eds.), *Handbook of Electric Motors* (2nd edn., pp. 547–616). New York: Marcel Dekker, Inc..
17. Pahl, G., and Rieg, F. (1984). Relative cost diagrams for purchased parts—Approximation polynomials aid in estimating costs. Konstruktion, 36, 1–6.
18. Palmgren, A. (1959). *Ball and Roller Bearing Engineering*. Philadelphia, PA: S. H. Burbank & Co., Inc.
19. Prasad, A. V. S. R. K., Rao, P. N., and Rao, U. R. K. (1997). Optimal selection of process parameters for turning operations in a CAPP system. International Journal of Production Research, 35(6), 1495–1522.
20. Salomone, T. A. (1995). *What Every Engineer Should Know About Concurrent Engineering*. New York: Marcel Dekker, Inc.
21. Shigley, J. E., and Mischke, C. R. (1996). Standard Handbook of Machine Design. The McGraw-Hill Companies New York, USA.
22. Winner, R. I., Pennell, J. P., Bertrand, H. E., and Slusarczuk, M. M. G. (1988). The role of concurrent engineering in weapons system acquisition. AD/A203615. IDA Report R-338. U.S. Department of Commerce: National Technical Information Service.

Chapter 3
Six Sigma: Continuous Improvement Toward Excellence

Qianmei Feng

Abstract Manufacturing and service organizations attempt to improve their products and processes by decreasing variation, because the competitive global market leaves little room for error. Variation is the biggest enemy of quality that is defined and evaluated by customers. The traditional concept of quality is based on average measures of the process/product and their deviation from the ideal target value. However, customers evaluate the quality of a process/product not only on the basis of the average but also by the variance in each transaction or use of the product. Customers want consistent, reliable, and predictable processes that deliver the best-in-class level of quality.

This is what the Six Sigma approach strives to achieve. Invented by Motorola in the 1980s, Six Sigma has been applied to many manufacturing companies, such as General Electric (GE), DuPont, and Ford. It has proven to be a customer-focused, data-driven, and robust methodology to improve the process and reduce costs. Over the last 20 years, Six Sigma has been successfully implemented in many industries, from large manufacturing to small businesses, from financial services and insurance industry to health-care systems. For example, under the partnership with GE, the Commonwealth Health Corporation launched the Six Sigma initiative in March 1998 and became the Six Sigma pioneer in the health-care industry. Six Sigma has been slowly but successfully implemented by many health-care institutions ever since.

3.1 What is Six Sigma?

As a data-driven and statistics-based approach, Six Sigma aims to deliver near-zero defects (as defined by customers) for the product, process, and transaction within an organization. The objective of using the Six Sigma approach is to reduce process

Assistant Professor, Department of Industrial Engineering University of Houston, Houston, Texas 77004

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

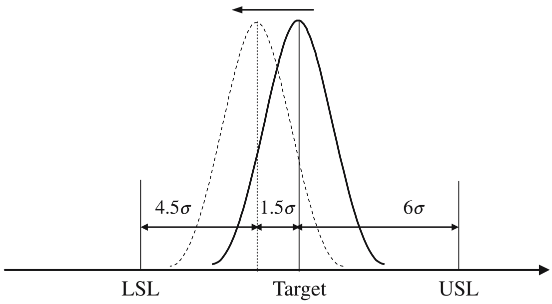


Fig. 3.1 A Six Sigma process with 1.5 σ shift

variation, so that the process results in no more than 3.4 defects per million opportunities (DPMO) in the long term. This defect rate is calculated on the basis of the assumption that many processes are prone to shift 1.50 standard deviations because of unavoidable assignable causes or degradation mechanisms. To achieve this long-term objective, the process capability has to reach the Six Sigma level in the short term, that is, the range between the target value and the specification limit contains six process standard deviations (6σ) on both sides of the target. In this way, the defect rate of a Six Sigma process is only about 0.002 DPMO. However, if the process mean shifts 1.5 process standard deviations over time, as shown in Fig. 3.1, the defect rate will increase from 0.002 DPMO to 3.4 DPMO.

Consequently, for a process that has a lower quality level than Six Sigma, the defect rate will increase significantly when the process shifts. A three-sigma process is used to be regarded as having good quality performance, before the introduction of Six Sigma. However, as shown in Fig. 3.2, the fraction outside of the specifications for the three-sigma process increases dramatically compared to the fraction for a Six Sigma process, which may cause serious quality problems over time. Therefore, three sigma is not good enough for many products or processes that attempt to avoid quality problems in the long run.

A Six Sigma process can also be interpreted in terms of process capability, which is associated with process variation [11]. The typical definition for process capability index, C_{pk}, is,

$$C_{pk} = \min\left\{\frac{\text{USL} - \hat{\mu}}{3\hat{\sigma}}, \frac{\hat{\mu} - LSL}{3\hat{\sigma}}\right\}$$

where USL is the upper specification limit, LSL is the lower specification limit, $\hat{\mu}$ is the point estimator of the mean, and $\hat{\sigma}$ is the point estimator of the standard deviation. If a process is centered at the middle of the specifications, which is also interpreted as the target value, then a Six Sigma process will have a capability of 2, that is, $C_{pk} = 2$. If a process wants to achieve 3.4 DPMO, it implies that the realized C_{pk} is 1.5 after the process shifts 1.5 standard deviations over time [6].

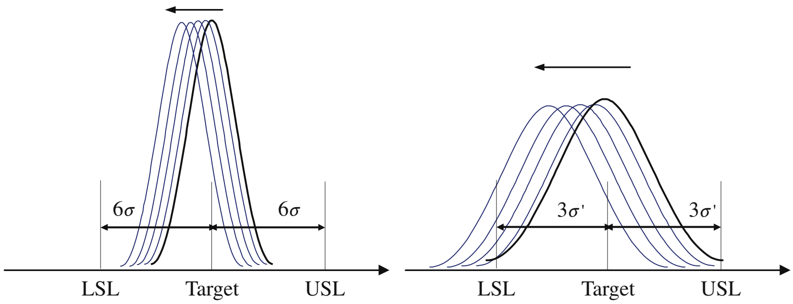


Fig. 3.2 Shifting a Six Sigma process and a three-sigma process

The requirement of 3.4 DPMO or C_{pk} of 1.5 is not the ultimate goal of Six Sigma. The goal is to establish the right business strategy toward organizational excellence. Six Sigma has evolved from a quality metric to an overall business management process that provides tangible business results to the bottom line through continuous process improvement. This is achieved by a dedicated workforce trained in Six Sigma methodology, on the project-by-project team basis, and with intensive support from the top management. As a customer-focused business strategy, Six Sigma puts customer satisfaction as the top priority: projects are driven and selected from the customer perspective, and performance standards are set on the basis of customer requirement. The Six Sigma methodology provides a road map for organizations to achieve the best-in-class business performance benchmarks.

Although the name does not contain the word "quality" or "performance," Six Sigma is a methodology for structured and process-oriented quality or performance improvement. To this end, two different yet similar road maps are available for organizations to choose from: one is the road map for Six Sigma process improvement, and the other is for "Design for Six Sigma" (DFSS). The DMAIC (Define, Measure, Analyze, Improve, and Control) methodology that most people are familiar with is the road map for Six Sigma process improvement, which involves the improvement of existing processes without changing the fundamental structure. DFSS is a Six Sigma approach that involves changing or redesigning the process at the early stages of product/process life cycle [16].

3.2 Why Six Sigma?

By identifying root causes and eliminating variations and defects, Six Sigma positively impacts many Critical-to-Quality (CTQ) features: timeliness/speed, cost, and quality of product/service. As shown in Fig. 3.3, these benefits will ultimately result in enhanced customer satisfaction, increased return-on-investment, and increased market share in the competitive global market.

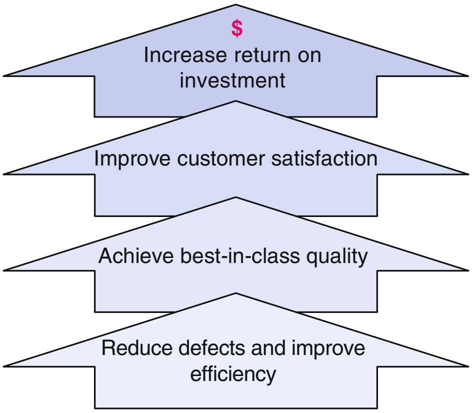


Fig. 3.3 Why Six Sigma? [3]

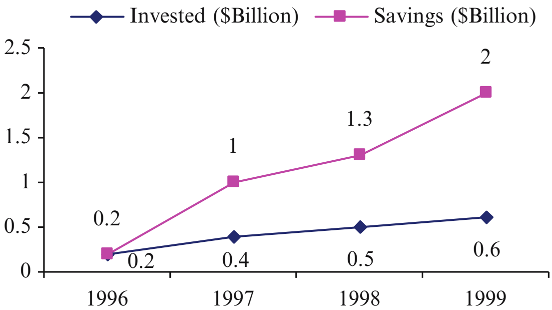


Fig. 3.4 Six Sigma progress at General Electric (GE) [15]

Six Sigma has helped many organizations to gain competitive advantages. For example, General Electric (GE) had more than$2 billion in savings during the first 4 years (1996–1999) of Six Sigma initiative, as shown in Fig. 3.4 [15]. These savings came from reduced rework, less waste, and decreased customer returns. Moreover, 53% of the Fortune 500 companies have deployed Six Sigma to some degree. Between 1987 and 2005, Six Sigma has saved them over $400 billion in total [12]. These convincing numbers speak for themselves regarding why Six Sigma should be implemented.

As the latest name for a comprehensive set of fundamental concepts, philosophies, tools, and methods for process or quality improvement, Six Sigma continues to show its endurance and return-on-investment for more than 20 years. By utilizing statistical tools, the Six Sigma methodology enables practitioners to accurately identify process hindering problems and demonstrate the improvements using objective data. However, Six Sigma is not just a collection of tools. The primary reason for the success of Six Sigma is that it provides a systematic approach for quality and process improvement. During most quality training in academia, industry, and government, students and professionals usually are taught a number of individual tools such as cause-and-effect diagram, statistical process control (SPC), experimental design, quality function deployment (QFD), and failure mode and effects analysis (FMEA), and leave the course without a mental big picture about how all these tools fit together. While implementing project-by-project, Six Sigma provides an overall process of improvement that clearly shows how to link and sequence individual tools. With Six Sigma, students and professionals know what to do when facing a real problem.

3.3 How is Six Sigma implemented?

As in many other management processes, the Six Sigma initiative may encounter more or less resistance from organizational members and executives. Naturally, members of any organization prefer to stay on the track where they are headed, unless some external force impels them to do otherwise. Many reasons lead people to resist changes, such as feeling “safer” to stick with the expectable status quo, fear of failures or breaking habits, and so on. Many factors must be considered to overcome these resistances. The key to implement Six Sigma successfully includes aligning critical success factors, building up effective deployment teams, and selecting the right projects to pursue.

3.3.1 Critical Success Factors

The expected benefits from implementing Six Sigma are driven by many factors, including organizational consistency of purpose, executive engagement and involvement, communications, and project successes. *Consistency of purpose* for quality and productivity improvement is the key for the principle-centered quality management in the twenty-first century and beyond [6]. Instead of alternating the purpose of quality management according to the trendy quality program, an organization should have the consistency of purpose for the principle-centered quality management.

Executive engagement is one of the most critical factors for Six Sigma to succeed. The consistent support and “buy in” from management are essential elements during the cultural change of implementing Six Sigma. As indicated in the survey of Six Sigma in US health-care organizations [7], lack of commitment from leadership is the major resistance/barrier for the successful implementation of Six Sigma. Executive engagement may include the following:

- Deploying Six Sigma as a core business process;
- Creating accountabilities and rewards system;

- Attending regular meetings to verify progress; and
- Commitment of time, resources, and people.

Effective communications on Six Sigma play an important role in creating the Six Sigma community within an organization. Communications may take the following forms [8]:

- Regular written communications on Six Sigma news and successes;
- Developing and disseminating communication aids to management;
- Advocating and creating a "common language" based on Six Sigma; and
- Communicating pertinent facts about Six Sigma in every company meeting.

One of the major differences between Six Sigma and other quality initiatives is its project-by-project way of implementation. The importance of *project successes* cannot be overemphasized for the Six Sigma implementation, especially at the initial stage. The project successes will bring benefits to business in a short time period (3–6 months), practitioners will feel satisfied for making improvements, and executives will see the benefits and provide full business buy-in. In this sense, selecting the right project can have a tremendous effect on laying the foundation for the success of Six Sigma.

3.3.2 Deployment Roles and Training

The implementation of Six Sigma starts with both the training of a dedicated workforce and the education across the organization. Although Six Sigma is deployed from top down, people in the organization need necessary training to understand the Six Sigma improvement and its potential benefits to the organization and themselves. A well-structured project team is one of the many advantages Six Sigma has over other quality programs. The team members are well-trained and certified at different levels of Six Sigma practice, so that they can work effectively in a team. The typical roles in Six Sigma belts structure include Champions, Master Black Belts (MBB), Black Belts (BB), and Green Belts (GB) [14].

Champion is an important role that bridges the operational level and the strategic level in Six Sigma projects. Champions are responsible to ensure that Six Sigma projects at the operational level are aligned with the strategic level business objectives. They need to provide BB the freedom to focus on the problem by keeping a BB from confrontation with executive managers. In addition to removing roadblocks, Champions should select projects accurately, adjust the speed of the deployment as necessary, and take responsibility for implementation [2].

MBB are typically hired as the leaders of Six Sigma teams who work closely with process owners. They are responsible for training BB and GB, providing technical expertise, and selecting appropriate projects if there are no Champions in the company. MBB are typically trained from BB who have demonstrated capability for solving difficult projects. Additional training is intended to broaden the tool sets and provide MBB with a wider array of skill sets.

At the operational level, BB are the "change agents" who are the heart and soul of the Six Sigma program. As full-time positions, BB help GB and other team members to understand the project and provide appropriate statistical techniques. BB are trained in the basic problem-solving tool and strategy and they are supported by MBB.

GB are employees trained by BB and/or MBB. They typically spend part time completing Six Sigma projects while maintaining their regular full-time work. As the project is completed, GB bring their experience in Six Sigma back to their regular work and begin to include the Six Sigma methodology in their daily activities. Thus, in the long run, GB are the ones who shift the culture of an organization [2].

3.4 Six Sigma Process Improvement—The DMAIC(T) Process

The benefits of Six Sigma process improvement are achieved through the utilization of a systematic approach, the DMAIC process. We extend it to a six-phase process, DMAIC(T), in order to emphasise the importance of technology transfer (T) of successful experiences [5]. Ideas or experiences can be transferred to similar products, processes, or transactions within an organization via an Intranet database of past Six Sigma projects. In this way, the rate of return on the investment from one Six Sigma project can be maximized [10, 11].

Six Sigma projects stay on track by establishing deliverables and reducing the process variation at each phase of the DMAIC(T) process. Each of the six phases answers the targeted question, which improves the effectiveness of the methodology continuously [5].

- **Define**—What is the problem that needs to be solved?
- **Measure**—What is the current process capability?
- **Analyze**—What are the root causes of the process variability?
- **Improve**—How to eliminate defects and improve the process capability?
- **Control**—What should be put in place to sustain the improvement?
- **Technology transfer**—Where else can the experience and/or lessons be applied?

The above questions are answered in each step sequentially as shown in Fig. 3.5. During the DMAIC(T) process, a practice problem identified by process owners, Champions, and/or MBB is translated to a statistical problem, which is solved by BB and/or GB. The statistical solution found is then transformed to a practical solution that can be implemented by the process owners. As shown in Fig. 3.5, the focus of Six Sigma shifts sequentially from monetary measures, to the process output variable (Y), to the process input variables or root causes (X_1, X_2, …, X_n), to the vital few input variables (X_i) that have critical impact on the output variable, and finally to the estimate of savings in money. The six phases are described in the following sections, and the key tools involved in each step are introduced as well.

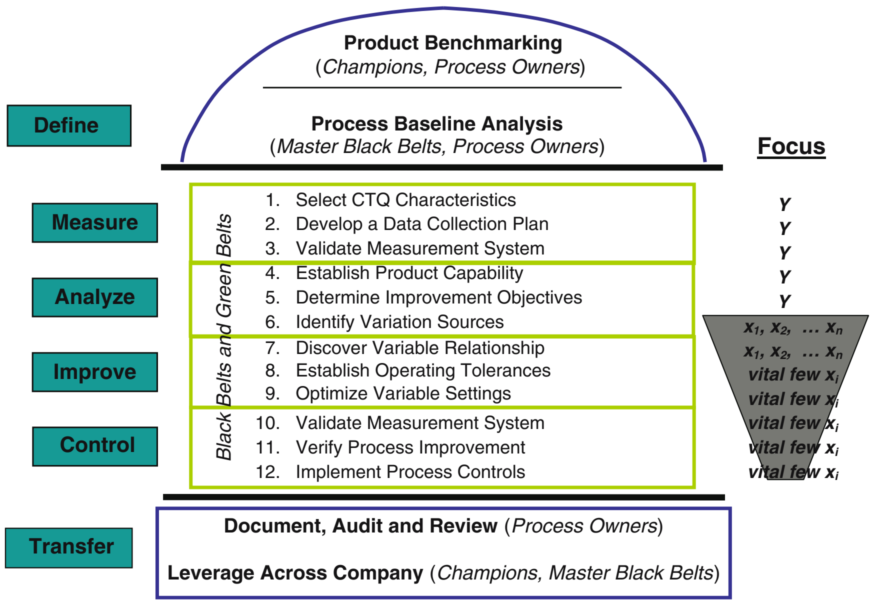


Fig. 3.5 The DMAIC(T) process

3.4.1 The DMAIC(T) Process

3.4.1.1 Phase 0: Define (D)

In the define phase, process owners, Champions, and/or MBB work together to identify the problem, define the project objectives, outline the expected benefits, form the team structure, and schedule the project timeline. Specifically, a project charter needs to be developed, including project scope, expectations, resources, milestones, and the core processes.

Six Sigma projects are customer-oriented as shown in Fig. 3.6. Based on customer requirements, the problem is identified and the project goals and deliverables are defined. Methods such as benchmarking surveys, spider charts, customer needs mapping, and SIPOC (supplier, input, process, output, and customer) **diagram** can be used to ensure that the customer requirements are properly identified. A general SIPOC diagram is given in Fig. 3.6 to illustrate the role of Six Sigma in a process. The critical to quality (CTQ) characteristics are defined from the viewpoint of customers, which are also called external CTQs. In the measure phase, the external CTQs are then translated into internal CTQs that are key process output variables (KPOVs).

3.4.1.2 Phase 1: Measure (M)

Six Sigma is a data-driven approach that requires quantifying the process using actual data. In this phase, the performance of the CTQ characteristics is evaluated

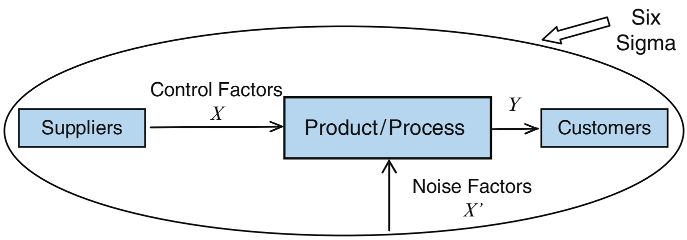


Fig. 3.6 SIPOC Diagram and Six Sigma

and compared to the customer requirements. The shortfalls are identified and the achievable opportunities are assessed.
Step 1.1: Select critical to quality characteristics

This step translates the customer requirements or external CTQs established in the define phase into internal CTQs or KPOVs denoted by *Y*. The performance of the process to be improved is often measured by one or a few KPOVs, which should be at the level the BB can impact. Fishbone chart, cause–effect matrix, QFD, and FMEA can be constructed to help the selection of KPOVs. The deliverables in this step include

- The selected KPOVs or *Y*s
- The identified defect or the undesirable outcome for each *Y*

Step 1.2: Develop a data collection plan

This step develops a data collection plan that gathers historical data over a business cycle, if possible. Six Sigma can then quantify the process using actual data.
Step 1.3: Validate measurement system

The capability of measurement systems needs to be evaluated to capture the variations due to sampling, operator, equipment, and ambient conditions. The repeatability and reproducibility of measurement systems can be assessed using Gage R&R or Analysis of Variance (ANOVA). This evaluation provides the decomposition of the total variability into components, and thus into targeted improvement actions.

3.4.1.3 Phase 2: Analyze (A)

Once the project is understood and the baseline performance is documented, it is time to analyze the process and find the root causes of problems using statistical tools. The objective is to understand the process in sufficient detail so that we are able to formulate options for improvement. To achieve this objective, the focus of the Six Sigma project will shift from the output variables to the process input variables in the analyze phase as shown in Fig. 3.7. The process input variables that

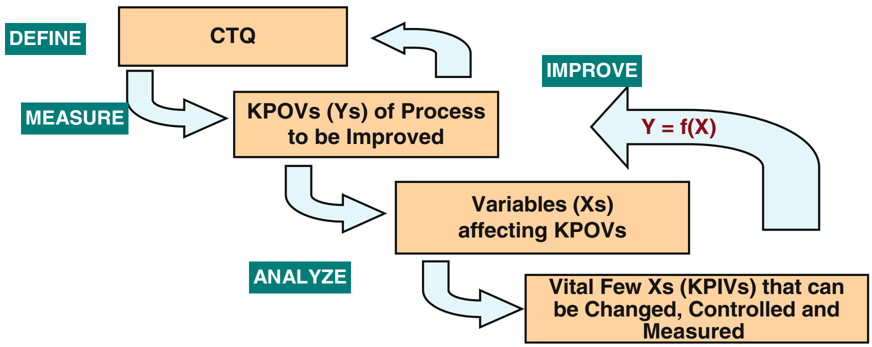


Fig. 3.7 Focus of Six Sigma

will lower the defect rates on KPOVs to achieve the project objectives are identified.

Step 2.1: Establish process capability

This step determines the current product capability, associated confidence levels, and sample size using the process capability analysis. The process capability index is closely related to the Sigma level of a process. For example, a Six Sigma process has the potential process capability of $C_{pk} = 2$ in the short term or when the process has no shift. The assessment of the current process behavior is obtained by analyzing historical data. The references to the current performance will provide an insight into achievable opportunities. The deliverables in this step include the defect rate in DPMO, Sigma level, and the process capability for each KPOV.

Step 2.2: Determine improvement objectives

The "best" Sigma level is defined for the project indicating the attainable process performance. The performance objectives are defined to establish a balance between improving customer satisfaction and available resources. The deliverables in this step include the achievable performance or benchmark for each KPOV.

Step 2.3: Identify variation sources

Starting from this step, the focus of Six Sigma shifts to the process input variables or *X*s, which include the controllable and uncontrollable factors, procedures, and conditions that affect the output variables. Some of these input variables will be used to control the output variables *Y*s, and these are called key process input variables (KPIVs) or vital few *X*s. The deliverables in the step include

- A list of all potential KPIVs that could impact the defect rates of KPOVs
- Identification of vital few *X*s

The key tools to search for the vital few *X*s include both graphical and analytical tools. The graphical tools are histograms, fishbone charts, Pareto charts, scatter plots, box plots, and residual plots. The analytical techniques include hypothesis testing (*t*-test, *F*-test, etc.), regression analysis, design of experiments (DOE), and SPC control charts.

3.4.1.4 Phase 3: Improve (I)

In the improvement phase, ideas and solutions are identified and implemented to initialize the change in the vital few *X*s. Experiments are designed and analyzed to find the best solution using statistical and optimization approaches.
Step 3.1: Discover variable relationships

This step explores the function relationship between the vital few *X*s and the *Y*s. Sometimes, the relationship is obvious if only one or two *X*s are identified. When many *X*s are present, it may be challenging to understand how these *X*s affect the *Y*s. A system transfer function (STF) can be developed as an empirical model relating *Y*s and the vital few *X*s [3]. The key tool to quantify the relationship is experimental design, which provides better understanding of the process than do the old-fashioned approaches, such as trial and error or changing one factor at a time.

Step 3.2: Establish operating tolerances

With the understanding of the functional relationship between the vital few *X*s and the *Y*s, we need to establish the operating tolerances of *X*s that optimize the performance of *Y*s. Mathematically, a variance transmission equation (VTE) can be developed that transfers variances of the vital few *X*s to variances of *Y*s [3].

Step 3.3: Optimize variable settings

The STF and VTE will be used to determine the key operating parameters and tolerances to achieve the desired performance of the *Y*s. Optimization models are developed to determine the optimum values for both means and variances for these vital *X*s. In this way, the changes in the *X*s are implemented.

3.4.1.5 Phase 4: Control (C)

The key to the overall success of Six Sigma methodology is its sustainability. In the control phase, the process improvement needs to be verified, and performance tracking mechanisms and measurements should be put into place to ensure that the process remains on the new course.
Step 4.1: Validate measurement system

The optimal settings for the vital few *X*s that optimize the output performance have been determined in the improve phase. To ensure that the optimal settings are achieved, the measurement systems need to be validated for the vital few *X*s, as applied in Step 1.3.

Step 4.2: Verify the process improvement

With the changes implemented in the vital few *X*s, the new process output variables need to be evaluated to verify the improvement. It involves the calculation of average, standard deviation, DPMO, Sigma level, and/or process capability for each KPOV. It may be necessary to check if the change in the process average (variance) is statistically significant before/after the improvement. Finally, we need to assess if the performance benchmark defined in Step 2.2 is achieved.

Step 4.3: Implement process controls

Controls need to be implemented to hold the gains, which involves monitoring the performance, developing corrective procedures, training people who run the process, and integrating into the systems. SPC is the major tool used to control the vital few *X*s and the KPOVs. The project is not complete until the changes are documented in the appropriate quality management system, such as QS9000/ISO9000. BB and process owners should work together to establish a turnover plan.

3.4.1.6 Phase ∝: Technology transfer

The infinity sign means that transferring technology is a never-ending phase for achieving Six Sigma quality. Ideas and knowledge developed in one part of the organization can be transferred to other parts of the organization. In addition, the methods and solutions developed for one product or process can be applied to other similar products or processes. The technology transfer can be implemented by creating a database of completed and ongoing Six Sigma projects that can be shared across the organization using the intranet. With the technology transfer, the Six Sigma approach starts to create phenomenal returns.

3.4.2 The Toolbox for the DMAIC(T) Process

Most of existing tools in the Six Sigma methodology are quality management tools and statistical methods, which is quite natural because Six Sigma originated from the statistical concept for quality improvement [1]. Typical quality management tools are process mapping, cause-and-effect diagrams, Pareto charts, QFD, FMEA, and so on. Examples of the statistical methods include the SPC, DOE, ANOVA, hypothesis testing, regression analysis, and so on [9]. These quality management and statistical tools are effective in finding and eliminating causes of defects in business processes by focusing on the inputs, the outputs, and/or the relationship between inputs and outputs. One of the advantages of the Six Sigma methodology over other process improvement programs is that the use of data analysis tools in Six Sigma projects enables practitioners to accurately identify process hindering problems and demonstrate the improvements using objective data. Table 3.1 summarizes the primary tools in the Six Sigma toolbox.

As experience in implementing Six Sigma accumulated, researchers and practitioners observed that Six Sigma has its inherent limitations and cannot be used as a universal solution for any process in any organization [13]. Therefore, additional techniques should be integrated to enhance the effectiveness of Six Sigma. Recent technical development in the field of management science and statistical analysis has provided more effective tools for improving the efficiency and the productivity

Table 3.1 Six Sigma toolbox

Phase	*Steps*	*Primary tools*
Define	Outline project objectives, expected benefits, and project timeline	Project charter Benchmarking surveys Spider charts Flowcharts SIPOC diagrams
	Select CTQ characteristics	QFD FMEA
	Develop a data collection plan	Sampling (data quantity and quality)
Measure	Validate the measurement system	Gage R&R ANOVA
	Establish product capability	Process capability analysis Basic graphical/summary statistics
	Determine improvement objectives	Cost analysis Forecasting
Analyze	Identify variation sources	Histogram/Pareto chart/Run chart Scatter plot Cause-and-effect diagram FMEA Hypothesis testing Confidence intervals ANOVA
	Discover variable relationship	ANOVA Linear regression Empirical modeling
Improve	Establish operating tolerances	Robust design ANOVA
	Optimize variable settings	Optimization techniques Sensitivity analysis Simulation
Control	Validate the measurement system	Gage R&R Process capability analysis
	Verify process improvement	Hypothesis testing
	Implement process controls	Statistical process control Control charts QS9000/ISO9000
Technology Transfer	Transfer solutions across the organization	Project management tools Database and intranet

SIPOC supplier, input, process, output, and customer *CTQ* critical to quality, *QFD* quality function deployment, *FMEA* failure mode and effects analysis, *ANOVA* **analysis of variance** of organizations, such as queuing systems, heuristics, and data envelopment analysis (DEA). Interested readers are referred to Tang et al. [13] and Feng and Antony [4].

3.5 Design for Six Sigma

While Six Sigma's DMAIC approach improves the existing process by removing defects, the fundamental structure of the process remains unchanged. To prevent quality problems from the beginning, the method of DFSS is more proactive, which involves changing or redesigning the process at the early stages of the product/process life cycle. The objective of DFSS is to "design it right at the first time" to avoid the quality defects downstream. Although DFSS takes more effort at the beginning, it will benefit an organization in the long run by designing Six Sigma quality into products/processes.

As shown in Fig. 3.8, the Six Sigma process improvement approach is effective in achieving the benchmark identified for the current process. To reach the future potential and make breakthrough, DFSS comes into play to redesign the existing process, or design a new process or product. DFSS becomes necessary when [16]

- the current process has to be replaced, rather than repaired or just improved,
- the required quality level cannot be achieved by just improving an existing process,
- an opportunity is identified to offer a new process, and
- breakthrough and new disruptive technologies becomes available.

DFSS is a disciplined approach to design a process or product that utilizes statistical tools and engineering methods. There are several methodologies for DFSS, such as DMADV, IDOV, or ICOV. The IDOV (or ICOV) acronym is defined as Identify, Design (Characterize the design), Optimize, and Validate, which is a well-known design methodology, especially in the manufacturing world. The DMADV is a popular methodology since it has the same number of

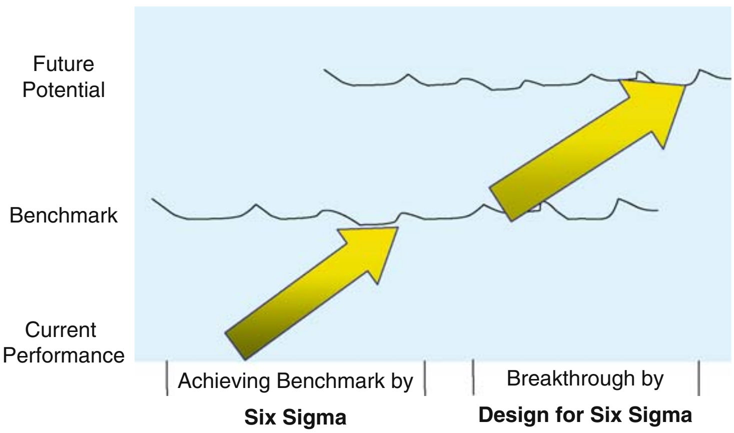


Fig. 3.8 Performance improvement by Six Sigma and DFSS

Table 3.2 Design for Six Sigma toolbox

Phase	Steps	Primary tools
Define	Define project goals and customer (internal and external) requirements	Project charter Benchmarking surveys SIPOC diagrams VOC (voice of customer)
Measure	Measure and determine technical requirements and specifications	QFD FMEA
Analyze	Analyze the process options to meet the customer needs	FMEA Risk assessment Engineering analysis
Design	Design the process to meet the customer needs	Robust design DOE Optimization techniques System engineering Simulation Statistical tolerancing
Verify	Verify and test the design, assess performance and ability to meet customer needs	Reliability testing Accelerated testing FMEA

SIPOC supplier, input, process, output, and customer, *VOC* voice of customer, *QFD* quality function deployment, *FMEA* failure mode and effects analysis, *DOE* design of experiment

letters as the DMAIC acronym. The five phases of DMADV are Define, Measure, Analyze, Design, and Verify. These are described in Table 3.2, as well as the typical tools for each phase.

DFSS integrates many well-known methods, tools, and philosophies for quality and reliability improvement, research, development and design strategies, and management thinking for teamwork from cradle to grave for products and processes in organizations. As pointed out in Welch et al. [15]: "Every new GE product and service in the future will be Designed for Six Sigma. These new offerings will truly take us to a new definition of World Class."

3.6 Case Study

A Six Sigma project implemented to measure physician productivity in a clinical department is given as a case study [4]. The concept of physician productivity is increasingly attracting attention in health-care sectors. As the result and measure of physicians' work, physician productivity can be used as the base of compensation assignment, resource allocation, and work incentive. One of the traditional measures of productivity is the number and types of patient encounters. However, without considering the time and other inputs invested in patient care, this measure does not allow for the measurement of physician efficiency. The application of traditional measures is limited by the complexity which exists in the actual practice.

3.6.1 Define Phase

A project aim statement that specified project objectives, expected benefits, team structure, and project timeline was created. The long-term goal of this study was to provide recommendations to the leadership that would lead to optimized resource planning and enhanced revenue. The project aimed to improve clinical productivity through the measurement of individual faculty productivity relative to the benchmark via persuasive and unique tactics.

3.6.2 Measure Phase

Physicians in the department differ in ages, experiences, and percentage of clinical times. In this study of assessing clinical productivity, three primary types of clinical outputs that capture the major contribution by most clinical physicians were considered: new outpatient visits, consulting, and established outpatient visits, which are the CTQ characteristics. This consideration was validated with the medical director in the department. The data could be collected from *a list of physician workload* (percentage of research, clinical, and teaching) and the *outpatient activity report* provided by the department.

3.6.3 Analyze Phase

The team analyzed the number and types of patient encounters using run charts and bar charts to provide illustration for the relative performance of individual physicians. The summary statistics were analyzed including mean, maximum, minimum, and standard deviation.

Furthermore, the following issues needed to be handled effectively:

- Multiple outputs including the numbers of new patients, consulting, and established patients shall be integrated to measure the overall productivity.
- Clinical inputs shall be incorporated into the evaluation to compare the relative efficiency.
- An easy-to-use efficiency measure is required in practice.

A fishbone diagram was constructed to analyze factors that influence physician productivity. Many factors can affect an individual physician's productivity, including patient characteristics, physician characteristics, hospital environment, and third-party reimbursement. It is reasonable to assume that the clinical cost can be estimated by the multiplication of physician's monthly salary and percentage of clinical time. This measure of clinical cost not only captures the budgetary input allocated to each physician but also reflects the clinical efforts produced by each physician. As the amount of physician's salary is closely related to physician's medical experience, age, and/or specialty, the clinical cost also conveys information about other characteristics of a physician. Therefore, a one-input and three-output clinical production system was in consideration.

3.6.4 Improve Phase

The DEA is an effective method to evaluate the relative efficiency among different organizational units. The DEA was implemented for the inputs and outputs, and the results provided efficiency ranking among physicians. The DEA yields additional information that can be used during the *Improve* phase. This includes the reference set consisting of efficient physicians for each inefficient physician as well as the performance levels that would make a relatively inefficient physician efficient. Beyond the recognition of inefficient physicians, the above-mentioned information can provide a countermeasure to improve physician productivity and optimize resource planning.

3.6.5 Control Phase

Using performance standards set by the DEA model, the relative efficiency for each physician can be monitored monthly. By collecting future data, cost savings can be analyzed to verify the benefits of implementing the DEA in the Six Sigma project. If the performance target is achieved for each physician, the overall efficiency can be improved, which will ultimately enhance organizational revenue with the same amount of inputs.

3.6.6 Technology Transfer Phase

The success of this project will buy-in extensive support from the leadership. The experience can then be transferred to other clinical departments in the organization for evaluating physician productivity and optimizing resource planning.

3.7 Conclusion and Future Trends

Although Six Sigma originated in the manufacturing industry, it has been successfully adopted by many other public or private sectors, from financial services to health-care delivery and management, from information technology to knowledge management. The successful implementation over 20 years supports the hypothesis that basic thinking and methods that are used in Six Sigma have lasting values, even though they may be marketed by new names in the future. Ideas can be integrated with other productivity improvement methods, for example, the recent focus on Lean Six Sigma. The methodology will continue to show their endurance in the global business environment.

References

1. Allen TT (2006). *Introduction to Engineering Statistics and Six Sigma: Statistical Quality Control and Design of Experiments and Systems*. Springer-Verlag, London.
2. Carnell M, Shank S (2002). The Champion's role in successful Six Sigma deployments. Retrieved August 29, 2007, from http://www.isixsigma.com/library/content/c020422a.asp.
3. Feng Q (2005). *Integrated statistical and optimization strategies for the improvement of Six Sigma methodology*. PhD dissertation, University of Washington, Seattle, WA.
4. Feng Q, Antony J (2007). Integrating data envelopment analysis into Six Sigma methodology for measuring health service efficiency. Submitted to Health Service Research (under review).
5. Feng Q, Kapur KC (2007). Quality control. In: Ravindran AR (ed.), *Operations research and management science handbook*. CRC Press, Boca Raton, FL.
6. Feng Q, Kapur KC (2008). Quality engineering: Control, design and optimization. In: Misra KB (ed.), *Handbook of performability engineering*. Springer-Verlag, London.
7. Feng Q, Manuel CM (2008). Under the knife: A national survey of Six Sigma programs in U.S. healthcare organizations. International Journal of Health Care Quality Assurance 21(6).
8. Hayes BJ (2002). Six Sigma critical success factors. Retrieved August 28, 2007, from http://www.isixsigma.com/library/content/c020415a.asp
9. Henderson GR (2006). *Six Sigma quality improvement with MINITAB*. John Wiley and Sons, New York, NY.
10. Kapur KC, Feng Q (2005). Integrated optimization models and strategies for the improvement of the Six Sigma process. International Journal of Six Sigma and Competitive Advantage 1(2).
11. Kapur KC, Feng Q (2006). Statistical methods for product and process improvement. In: Pham H (ed.), *Springer handbook of engineering statistics*. Springer-Verlag, London.
12. Marx M (2007). Six Sigma saves a fortune. *iSixSigma Magazine*, January 2007.
13. Tang LC, Goh TN, Lam SW, Zhang CW (2007). Fortification of Six Sigma: Expanding the DMAIC toolset. Quality and Reliability Engineering International 23(1): 3-18.
14. Tsung F (2006). Six Sigma. In: Pham H (ed.), *Springer handbook of engineering statistics*. Springer-Verlag, London.
15. Welch JF, Murphy EF, Dammerman DD, Opie JD (1999). Letter to our share owners. Available at: http://www.ge.com/annual98/share/index.htm.
16. Yang K, El-Haik B (2003). *Design for Six Sigma: A roadmap for product development*. McGraw-Hill, New York, NY.

Chapter 4
Supply Chain Workflow Modeling Using Ontologies

Charu Chandra

Abstract One of the primary objectives of supply chain (SC) information support system is to develop a conceptual design of organizational and process knowledge models which facilitate optimal SC management (SCM). The SC knowledge modeling consists of two components: (1) modeling SC workflows and (2) capturing and organizing knowledge necessary for managing them. Workflow modeling deals with handling activities to generate and utilize knowledge, whereas ontology engineering formalizes knowledge content. This chapter proposes a framework comprising both aspects of knowledge modeling. To model workflows, a combination of two frameworks is proposed: (1) SC operation reference model for higher-level process and (2) process modeling tools, such as integrated definition (IDEF) and unified modeling language (UML), for the lower, application-level process model representation. For workflow knowledge capturing and representation, two standards are introduced: situation calculus and SC markup language (SCML). The former is utilized for capturing process logic with mathematical expressions, and the latter for coding this logic with a computational language. An example of production scheduling for a steel SC is provided as an illustration.

4.1 Introduction

The problem of system analysis, modeling, and representation has always been important from the perspective of understanding the organization and its processes and supporting process management. A system model is a description of its constituents, namely, goals, processes, relationships between processes, and mechanisms for managing these processes. The fundamental question this chapter is seeking to address is, "how supply chain (SC) (as an organization) can be modeled to facilitate the development of supporting information system defined appropriately

Industrial and Manufacturing Systems Engineering Department, University of Michigan, Dearborn, Michigan 48128

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5, © Springer Science + Business Media, LLC 2008

for problem-solving methods." Toward this end, Crubezy and Musen [7] suggest knowledge management methodology based on ontologies. The rationale is to develop problem-solving methods as independent components for reuse in decision modeling application systems. This chapter adopts their approach and extends it to propose a framework, whereby problem-specific ontologies can be developed to conceptualize knowledge about SC processes and problems.

A holistic approach is applied to SC knowledge modeling, where SC is considered as a system consisting of processes interrelated to each other. Each process may have various views reflecting different system management perspectives. In order to design these process views, conceptual models are proposed, which are graphical representations used for symbolizing both static (things and their concepts) and dynamic phenomena (events and processes), respectively [34].

One of the primary objectives of SC ontology development is to develop a conceptual design of organizational and process knowledge models which facilitate optimal SC management (SCM). The SC knowledge modeling consists of two components: (1) modeling SC workflows and (2) capturing and organizing knowledge necessary for managing these workflows. For knowledge organization, this chapter suggests utilizing concepts from ontology engineering. Workflow modeling deals with handling activities to generate and utilize knowledge, whereas ontology engineering formalizes knowledge content. The majority of knowledge that needs to be managed in an organization is generated and utilized in-house along the workflow of organization [20]. Therefore, it is meaningful as well as useful to build the process knowledge content based on the structure and logic of processes and tasks defined in the workflow.

The rest of the chapter is organized as follows. Sect. 4.2 describes SC domain, and the impetus leading up to the development of ontology-based workflow models. Sect. 4.3 presents contemporary research in areas of workflow management, ontology engineering, and their intersection. Sect. 4.4 introduces components of proposed approach for workflow modeling and ontology engineering. Sect. 4.5 outlines a meta-model for tying framework components together. Sect. 4.6 introduces a new programming language: SC ontology language (SCOL), and its two constituents: extended situation calculus and SC markup language (SCML). Sect. 4.7 presents the application of proposed approach to a real-world situation.

4.2 Background and Motivation

The SC is a network of facilities wherein various distribution options and approaches are utilized to effectively integrate suppliers, manufacturers, and distributors through performing functions of procurement of materials, transformation of these materials into intermediate and finished products, and their distribution to customers in the right quantities, to the right locations, and at the right time, in order to meet required service levels with minimal cost.

Historically, the management of an organization had relied on well-established hierarchy. But modern enterprises, such as SC, are more involved in peer-to-peer relationships, and the necessity of horizontal collaboration overshadows the hierarchical management style. The distributed environment imposes new challenges for design and management of processes and tasks in SC. The communication of essential information and decisions is becoming critical in allowing distributed organization to operate in a cooperative manner, integrate cross-organizational functions, and manage processes jointly. In order to make these functions and processes streamlined, these have to be managed in a holistic and synergistic manner. The former considers SC as a whole consisting of parts, and the latter is a set of independent components collectively working on common problems. Centralized control is not very effective for managing SC. Fox et al. [12] propose agent-oriented SCM, where SC is viewed as being managed by a set of intelligent agents, each responsible for one or more tasks in SC, and interacting with other agents in the planning and execution of their responsibilities. Every agent requires specific knowledge for its operation. Ontologies provide the vocabulary for representing domain or problem knowledge and are also crucial for enabling knowledge-level interoperations of agents.

The role of system analysis and conceptual modeling is acknowledged as a powerful tool for understanding system processes, tasks, and activities [2]. Typically, conceptual models are required to

- understand the SC organization,
- document SC processes,
- specify functional and user requirements for a SC system,
- design information system to support processes, and
- evaluate the change in business process reengineering.

The necessity of workflow knowledge modeling arises on the basis of several reasons, such as when

- the extent of knowledge becomes intractable,
- business units are geographically decentralized, but more closely networked,
- collaboration becomes important among individual workers, and
- challenges are faced in eliciting requirements when user cohorts are large, decentralized, and unknown.

Research findings in various fields of study spur the development of knowledge-based frameworks for workflow modeling. These incentives can be specified as follows:

- The emergence of object-oriented approach,
- The use of process models,
- The potential of conceptual models to assist business process reengineering,
- Ontology application in domain and process knowledge representation, and
- Agent technology for automatic process management and control.

Object-oriented methodology suits the inherent structure of SC domain. In software development domain, the system consists of objects with properties and functionalities, whereas in SC modeling domain, the system consists of processes, which have objects. Process models explicitly document tasks and activities, thus ameliorating the complexity of the SC system. Studying separate process models and building knowledge for their management is much easier than to deal with the entire system with its uncertainty and dynamics. Conceptual models provide techniques for documenting and validating processes. Various SC alternatives can be modeled and validated before the best solution is chosen for a particular situation. Ontology provides a common view of situations and processes, thus ensuring shared understanding of common problems. This research initiative views SCM as a set of tasks and activities managed by software agents. These are intelligent goal-oriented software entities acting autonomously or semi-autonomously and collaborating with each other through messages. All these technologies provide a background for ontology-driven process modeling and management.

4.3 Literature Survey

The SC workflow knowledge modeling framework proposed in this chapter consists of two components—conceptual modeling of workflow and ontology engineering. The state of the current research in these two fields of study is examined and presented in this section.

4.3.1 Workflow Modeling

Workflow modeling is identified as the means to define and administer business processes automatically. Most of the research initiatives in this field are focused on information system design for implementing real-time collaboration systems and process automation [1, 8]. The workflow management coalition (WFMC) (www.wfms.org) has established standards for reference model to design workflow and specifications of data and documents to realize interpretability between workflow systems. According to WFMC, workflow is "the automation of a business process, in whole or in part, during which documents, information or tasks are passed from one participant to another for action according to a set of procedural rules." The automation of business processes increases efficiency, facilitates the creation of virtual organizations, and offers new potential for e-commerce solutions. The management and control of workflows is the task of workflow management system (WFMS). Process modeling is the most important component of workflow management. This and other three components, namely, goal, structure, and object views, are described in IBM [19]. This chapter considers only the process view of workflow management.

Process modeling enables systematic analysis of business, focusing on tasks (functions) that are performed regularly, controls to ensure their proper performance, resources needed to perform a task, results of a task, and inputs (raw materials) on which the task operates. Several research efforts have proposed methodologies to improve enterprise performance through modeling and design of business processes [30, 31]. The common thread in these approaches is the use of process models as an aid to understand and design systems. A process can be looked from different perspectives depending on the type of information required. Previous research has defined a number of views with corresponding methodologies, namely, integrated definition (IDEF), computer integrated manufacturing open systems architecture (CIM-OSA), architecture of integrated information system (ARIS), and Petri nets. Recently, with the emerging growth of object-oriented paradigms for analyzing and designing systems, unified modeling language (UML) is in use for business process design.

Workflow systems may play a significant role in SCM, especially when SC members are geographically distributed but are closely tied to business processes.

4.3.2 Ontology Engineering

Ontologies have shown their usefulness for various applications areas, such as knowledge representation, intelligent information integration, expert systems, and active database systems. Ontology refers to an engineering artifact, constituted by a specific vocabulary, used to describe a certain reality, in addition to a set of explicit assumptions regarding the intended meaning of words in the vocabulary [18]. Ontology has been applied for capturing the static nature of the system [27] and its dynamics, for which situation calculus is utilized [23].

Ontology has been found useful in modeling SC by developing knowledge bases specific to problem domains [4]. Ontology can also be used as means for bridging system analysis and application system constructions. A survey of literature reveals three dimensions of ontology engineering process [18, 33]. The first dimension is the building stage, consisting of several activities: specification, conceptualization, formalization, implementation, and maintenance. The second dimension is the type of ontology: domain and problem [17]. The third dimension is ontology modeling components. Chandra and Tumanyan [6] specify three components of SC ontology: (1) domain concepts with their relationships, (2) axioms for representing rules held on this domain, and (3) problem-solving algorithms, if it is problem ontology. This chapter does not intend to cover all these aspects. From the first dimension, we discuss specification, conceptualization, and formalization. From the second dimension, only problem ontology type is covered. As to ontology modeling dimension, all three components are demonstrated.

4.3.3 Knowledge-Intensive Workflow Management

This subsection demonstrates research works in the area of knowledge management, particularly ontology application to workflow management. Researchers trying to model workflows and make processes managed effectively are offering techniques borrowed from the knowledge management discipline. Casati et al. [3] propose automatic derivation techniques of active rules that may form workflow specification. For describing and representing these rules, active database systems have been utilized. Through active rules, workflow performance is represented with operational semantics. Workflow management using e-commerce is proposed by Basu and Kumar [1]. For modeling workflows, and controlling and monitoring their performances, organizational meta-models are proposed to be designed. Meta-models incorporate organizational constraints as they relate to resources, roles, tasks, policies, etc. Knowledge management technique is introduced in Kang et al. [20] to combine process and content management. Process management is concerned with handling activities to generate and utilize knowledge, whereas content management deals with the knowledge content.

4.4 Conceptual Framework

The literature survey presented in the previous section revealed the importance of knowledge-based techniques in workflow modeling and management. Most of the frameworks found in literature are devoted to describing organizational workflow structure [1, 32], or to presenting the usefulness of knowledge-based systems in managing workflows [3, 20]. There are few research works showing the mechanisms of analyzing processes, and modeling knowledge to support their functions. The framework presented herein intends to fill this gap. This chapter proposes an approach to SC system analysis and a conceptual model design in the form of ontologies. The framework consists of two major modeling environments: SC workflow modeling and ontology engineering (Fig. 4.1). The result of SC workflow modeling is a unified representation of SC processes in a hierarchy that identifies relationships among them. At this stage, processes are documented with explicit models. Ontology engineering deals with capturing knowledge for every process and task designed in the previous stage. As a result of application of these models, knowledge modules in the form of ontologies are developed.

4.4.1 Supply Chain Workflow Modeling

The SC workflow model is the collection of business processes, their relationships, and sets of characteristics, necessary for evaluating these processes. Two main components of the SC workflow model are distinguished, identifying two levels of

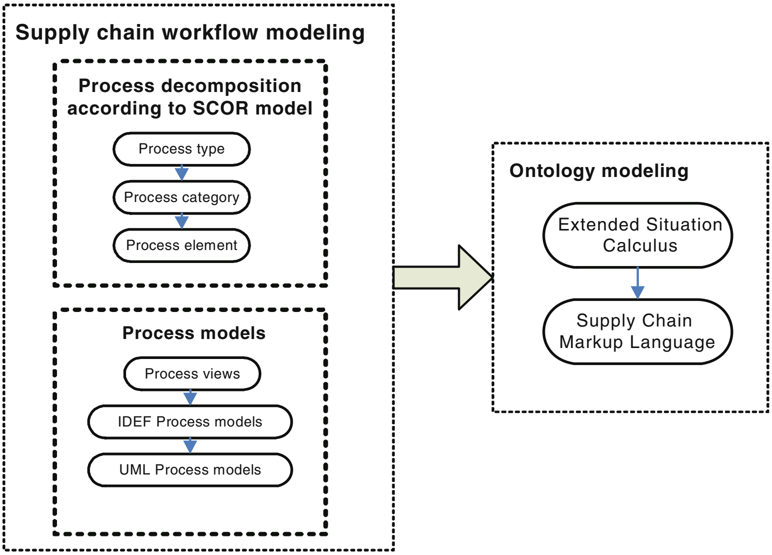


Fig. 4.1 Supply chain analysis and modeling conceptual framework

process representation abstraction. For capturing higher-level process, we have adopted the SC operations reference model (SCOR; www.supply-chain.org). For the lower level, process modeling tools such as IDEF and UML are utilized.

4.4.1.1 Supply Chain Operations Reference Model

SCOR integrates concepts of business processes, benchmarking, and best practices into a cross-functional framework. SCOR builds a hierarchy of SC processes, which can be divided into three levels: process type, process category, and process element. Process type defines five basic management processes in SC (Plan, Source, Make, Deliver, and Return) that provide the organizational structure of SCOR.

The second level defines three process categories: planning, execution, and enable. A planning element is a process that aligns expected resources to meet anticipated demands. Execution processes are triggered by planned or actual demand that changes the state of products. They include scheduling and sequencing, transforming materials and services, and moving product. Enable processes prepare, maintain, and manage information or relationships upon which planning and execution processes rely. The SCOR second level also defines criteria for process classification, for example, for Make process type, three categories are identified in SCOR: M1 make-to-stock, M2 make-to-order, and M3 engineer-to-order. The third level presents detailed process elements' information on each process category described at the second level, particularly process input, output, process flows, performance attributes, and best practices for their implementation.

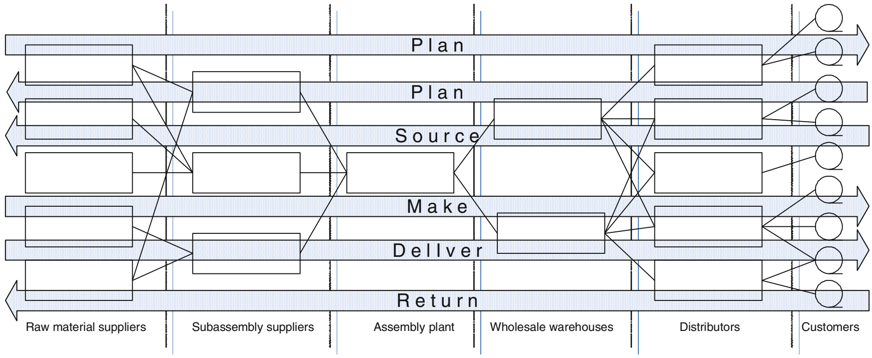


Fig. 4.2 Supply chain structure and supply chain operations reference-model (SCOR) processes

SCOR five processes residing at the first level as well as their subprocesses at the second and third levels are flows across SC and are depicted in Fig. 4.2. Processes categorized as planning are processes that balance aggregate demand and supply to develop a course of action which best meets sourcing, production, and delivery requirements. This process can flow both from left to right and from right to left. The Source process flows from right (distributors, wholesalers) to left (raw material suppliers) and is for procuring goods and services to meet planned or actual demand. The Make process flows from left to right, transforming input materials into products to a finished state. The Delivery process provides finished goods and services to the downstream tier and eventually to customers and includes order management, transportation management, and distribution management. The Return process is associated with returning or receiving returned products for any reason.

4.4.1.2 Process Modeling

SCOR does not provide mechanisms for detailed level process specifications. This level is proposed to model using a combined methodology, a best breed of IDEF and UML. Process modeling aims to represent processes specified in SCOR third level as a collection of tasks executed by various resources within an SC. Each process transforms a specific set of inputs into a specific set of outputs to achieve some functional goals. Kim et al. [*21*] in their study of WFMS suggests nested process modeling, where each business process can be broken down into subprocesses or tasks. A structure can be provided for hierarchically arranging them into taxonomy, making it easier to grasp the relationship between processes and tasks. In turn, tasks can be decomposed into activities yielding another level in problem taxonomy. SC has some specifics that cannot be adequately represented with WFMS, such as distributed nature, loose connection among SC members, and different process management standards. WFMC precisely defines the motivation, the usefulness, and the theoretical aspects of WFMS. In building SC workflow system, the framework proposed by WFMC will be used as a reference and as a testing tool for evaluating the correctness and usefulness of the proposed system. Since the reference model proposed by WFMC is becoming the standard for business process management, the framework for workflow modeling proposed in this chapter will be integrated with standards promoted by WFMC. Process (task, problem) representation is to be compatible with formalisms proposed by WFMC, and XML process definition language (XPDL). An XSL translator is to be developed for this purpose. It provides transformation languages which enable describing how files encoded in XML standard can be transformed.

The above-described features of workflow modeling can be captured by using explicit models. In comparing various business process modeling methods provided by Lin et al. [24], four methods have been selected: IDEF0, IDEF1, IDEF3, and object-oriented modeling with UML formalism. IDEF0 method is designed to model decisions, actions, and activities of an SC targeted to analyze its functional perspectives. IDEF1 is an information modeling method used in identifying (1) the information collected, stored, and managed by an SC; (2) rules governing the management of information; (3) logical relationships within enterprise reflected in information; and (4) problems resulting from lack of good information modeling [25]. IDEF3 describes processes as sequence of events and activities. It is a scenario-driven process modeling technique based on precedence and causal relationships between events and situations. IDEF3 model provides the method for expressing and documenting SC domain experts' knowledge about how a particular process works, in contrast to IDEF0, which is concerned with what activities an SC performs.

IDEF formalism documents processes with semantic diagrams, which is a part of workflow management. The other part is how to manage these processes intelligently by designing information system to support functions and activities. Transferring the business model into a software model is necessary to design adequate information system. An object-oriented process modeling technique is proposed to accomplish the transformation from business to software view. UML modeling formalism provides a unique opportunity in this respect. UML offers a library of diagrams to semantically present process views captured by IDEF formalism. UML meta-models define constructs that designers can use for modeling a software system. This chapter offers the best breed of these two techniques, whereby IDEF is utilized for handling low-level process granularity, and UML offers object representation and migration to software applications.

4.4.2 Ontology Engineering

Ontology development is the last stage of the SC conceptual modeling framework proposed in this chapter. Chandra and Tumanyan [6] have proposed ontology development as a backbone for SC information system design, where ontology participation in information system is described and ontology development stages are presented.

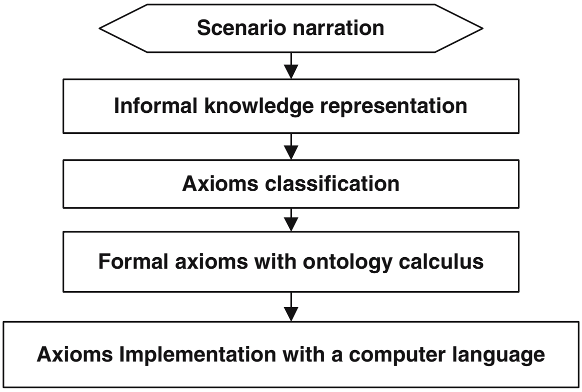


Fig. 4.3 Ontology engineering conceptual framework

The ontology-engineering framework presented in this chapter and depicted in Fig. 4.3 proposes two ontology development specifications from the perspectives of knowledge and software engineers, namely, situation and predicate calculus, and a new XML specification, called SCML.

Processes documented in workflow modeling environment are studied with domain experts and narrative description of the necessary knowledge is captured in a text or another word processing format. Scenario narration describes the case study in English and presents the situation to be modeled. It has a form of story problems, or examples not adequately addressed by existing information system. Scenario narration may also contain problem analysis and possible solutions to the described problem. Each process is considered as a stand-alone, conceptually closed system and studied in terms of identifying properties and rules relevant to them.

From narrated descriptions, informal knowledge representation captures questions that ontology must address. These are English descriptions presented in a modular way. Competency questions [16] can be considered as requirements that ontology is to address. Questions are mostly constraints on objects and activities, but they may also define preconditions that will trigger an action, and the sequence of activities, which necessarily should follow each other. These questions do not generate ontological commitments, since they are not formal representations, but can be used for evaluating the expressiveness of designed ontology.

Once knowledge about processes is presented in a modular form, it should be structured. Ideally axioms should be arranged in hierarchy with higher-level questions requiring the solution for lower level questions. It can be noticed here that knowledge classification is defined in workflow modeling stage. Ontology engineering inherits this structure (discussed in the next section) and accommodates identified questions into this structure. If it is necessary, especially when processes are complex and the number of questions is significant, sub-hierarchies inside each process can be constructed.

Recall, ontology is a formal specification of a conceptualization [14], hence we need a formalism to capture collected knowledge and present it in a language that is processable by machines and comprehensible by people. For formalizing informal knowledge, we need semantics for its expressiveness and terminology for implementing axioms in a language. Formal ontology axioms formulation can be accomplished through situation calculus [29]. Extending the standard terminology of situation calculus for capturing SC-specific predicates and situations will allow having a complete set of statements for expressing knowledge captured and documented in previous stages.

Knowledge represented with situation calculus is the mathematical model and needs to be coded in a computational language. Implementation of axioms is the coding process, where statements in situation calculus are represented with XML documents. Despite critics [9], we advocate the sufficiency of XML documents' capabilities for fully expressing the content and structure of explicit ontologies.

For each process item (process, task, or activity), ontology or a set of ontologies is designed. Ontologies conceptualize the knowledge necessary for planning and executing these process items. The knowledge encapsulated in ontologies consists of three components: data model, axioms defining constraints and rules held on data model, and algorithms, which are step-by-step conditional descriptions of process flows.

4.5 Supply Chain Knowledge Modeling: A Meta Model for Process Integration

The previous section outlines the components of SC knowledge modeling conceptual framework. However, it does not address the issue of mappings between process modeling and ontology engineering. SC integration can be facilitated if SC processes are integrated. The latter is possible if there is a common semantics of process information semantics, which can be implemented through ontologies. The development of ontologies is motivated by the necessity for providing integration of process models for delivering to them common semantics. Among existing semantic frameworks, Petri net can be considered as a potential competitor for ontologies [35]. However, there are no agreed-upon standards for semantics of Petri nets. In contrast, proposed ontology engineering aims to deliver common semantics to process models captured by a frame-based hierarchy (see next section) and a set of axioms.

The proposed process modeling framework can be represented as a hierarchy of SC processes, where higher levels are for representing more generic issues, such as planning and execution, and lower levels are for representing specific issues. The taxonomy of SC processes and the problems associated with them are depicted in Fig. 4.4.

Each level defines the level of information generalization that process models capture. Process models provide explicit representations, whose study may reveal

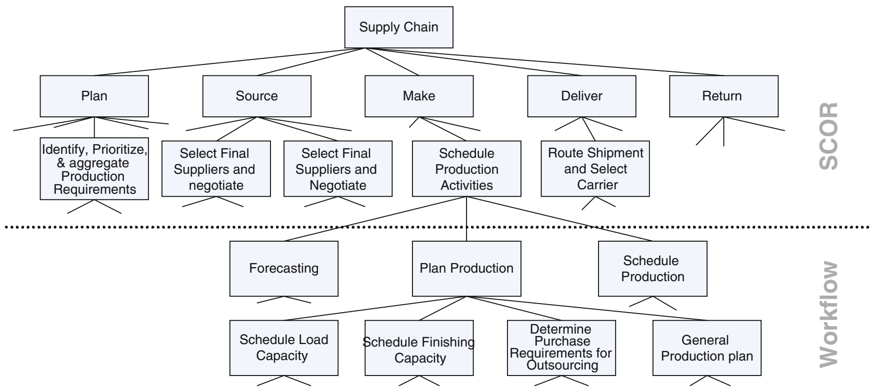


Fig. 4.4 Taxonomy of supply chain processes

requirements and questions that ontologies must be able to answer. Grüninger and Fox [15] call these questions as competencies for ontologies which should address the following issues:

- Entities and their properties
- Relationships inside the problem and with other problems
- Constraints
- Behaviors

Above-mentioned are general questions that ontologies should address. Specific competency questions may be relevant to individual levels, such as the process elements residing at the third level in problem taxonomy comprises issues related to best practices for solving these problems.

The structure depicted in Fig. 4.4 reflects the hierarchy of ontologies to be designed. SCOR defines all necessary properties for building ontologies for the first three levels. Ontologies design at these levels is not practical in terms of their usefulness and utilization by software applications, but having these ontologies in the first place provides to low-level ontology developers reusable knowledge modules that can be used for various applications. Mapping between process modeling and ontology development for two sub-layers is demonstrated below.

4.5.1 SCOR Ontology

The SCOR describes (1) activities in three generalization levels, (2) relationships between activities in terms of defining output parameters of some activities that serve as input parameters for others, (3) benchmarks for each activities, and (4) best practices for activities. SCOR ontology systematically documents these features with frame-based and first-order formalisms (these two frameworks will be discussed in the next section). For example, for "Schedule Production Activities" box

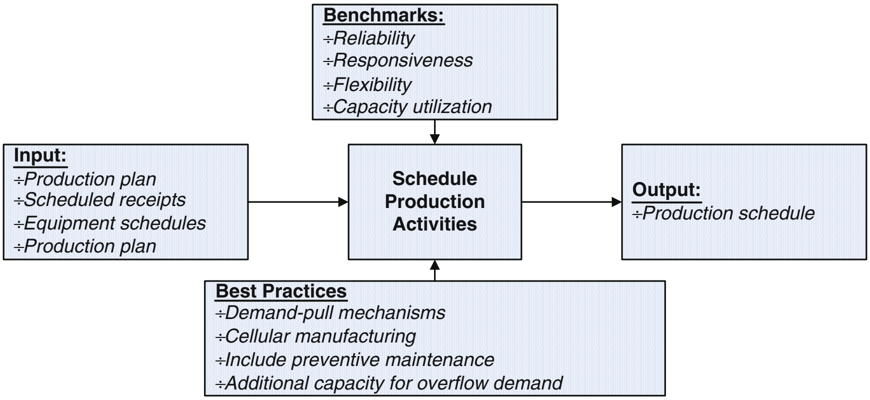


Fig. 4.5 Supply chain operations reference-model (SCOR) ontology elements (adapted from SCOR terminology)

(Fig. 4.4), the following information is to be captured and represented in ontology constructs: (1) input parameters, (2) output parameters, (3) benchmarks, and (4) best practices. SCOR defines and documents the information depicted in Fig. 4.5.

The only thing is to follow ontology language specifications for formalizing implicit knowledge provided by SCOR into a formal language.

4.5.2 Workflow Ontology

Ontology engineering for this level is much complex than for the previous level, since characteristics describing processes at various sublevels are not standardized and need to be defined in the first place. Knowledge conceptualized at this level is supposed to deliver to decision modeling applications knowledge necessary for building simulation, optimization, or other mathematical models to reason about the problem of interest. Upper level processes are for macro-focused decision modeling, lower levels are for micro-focused. As can be seen from Fig. 4.4, "forecasting" is a subprocess of "Schedule Production Activities" box from SCOR. Ontology engineering conceptual framework depicted in Fig. 4.3 is applied for each process. Scenario narration for forecasting is the description of (1) the type of demand, namely, stochastic or deterministic, seasonal or with distribution; (2) sales plan; (3) forecast method; and (4) plant capacity.

Informal knowledge representation deals with defining rules between concepts identified in the previous stage, such as how plant capacity is related to sales plan. These relationships are not as simple as they may seem at first glance. Plant capacity and sales plan may have many characteristics and they may be engaged in nonlinear relationships, which sometimes are impossible to model with mathematical expressions. Axiomatization of identified rules is a matter of making informal knowledge formal by describing them with algebraic expressions called situation calculus. The narrated description of one of the possible rules is “If the demand is seasonal, apply time-series forecasting method”; its algebraic representation will be

$$\text{Poss(Demand,Seasonal)} \equiv \text{Apply(Time\_Series,method)} \tag{4.1}$$

Situation calculus is utilized for formal representation of rules. These are documented process models that software engineers can translate into a computation language.

For the last two stages of ontology engineering (Fig. 4.3), namely, rules axiomatization and computational implementation, we need to define specifications, and the next section is devoted for this purpose.

4.6 Ontology Language: Introduction to a Specification

Ontology development requires semantics for building ontologies and expressing the informal knowledge into explicit ontologies, for which a programming language—SCOL is introduced. This language must meet six requirements:

1. It should be comprehensible to human users. Object-oriented modeling paradigm and frame-based logic satisfy this criteria.
2. It must be compatible with existing knowledge representation formalisms. Currently, XML and resource definition framework (RDF) are becoming the main languages for this pursuit.
3. It must have enough expressiveness for use in the Web environment.
4. It must have well-defined semantics for providing rich modeling primitives and efficient reasoning support.
5. It must have a good level of syntactical development to allow effective development of parsers.
6. It must be capable of expressing activities.

Many ontology languages are proposed by research community, such as Knowledge interchange format (KIF) [10] and ontology inference layer (OIL) [11]. None of these languages meet requirements identified above. KIF cannot be used in the Web environment. OIL is developed as a Web-based representation and inference layer for ontologies, which combines the widely used modeling primitives from frame-based languages with the formal semantics and reasoning services provided by description logics but fails to express activities.

The proposed SCOL is based on three fundamentals:

- Frame-based logic for providing modeling primitives.
- First-order logic for specifying the formal representation of competency questions related to SC activities and their reasoning.
- Language syntax for providing syntactical exchange notations in the Web environment.

4.6.1 Frame-Based Logic

Object-oriented approaches and frame-based systems provide similar views to the domain. Their central modeling primitive is the class (or frame), which has properties. Frame or class provides a certain context for modeling of one aspect of a domain. SCOL incorporates the modeling primitives of frame-based system into its language and process models represented with UML class diagrams constitute its semantics. Frame logic describes the knowledge in the form of concepts with subsumption relationships and attributes and role restrictions. It is utilized to express structured knowledge formulated as frame-based systems and classification taxonomies. For frame logic representation, description logic is proposed by Fensel et al. [11], on the basis of which OIL ontology language is introduced. This chapter advocates the use of UML class diagram for expressing frame-based logic. Being semantically rich, class diagram possesses a set of primitives necessary for representing concepts with their subsumption relationships. On the contrary, UML software applications support translation function of their diagrams into XML documents, which completely fits the intentions of this research effort in employing XML as formalism for SCOL computational implementation.

SCOL inherits its semantics for expressing taxonomies and added new primitives necessary for capturing SC activities from UML class diagram. For this purpose first-order logic is utilized.

4.6.2 First-Order Logic

First-order logic explicitly models SC domain changes as the result of performing actions. It also defines predicates identifying when and how action should take place. There are varieties of ways for modeling activities. Situation calculus as a tool for first-order logic representation has been adopted to provide semantics to SC ontology of system activity and state through axioms. In situation calculus, the sequence of actions is represented with first-order term called *situation*. These representations are carried out with the help of symbols denoting an action or a predicate. Thus, the symbol Do(x,s) represents a new state s_1 which is a result of an action applied in situation s. Situations, whose true values vary from situation to situation, are called functional *fluents*. These are taking the situation s as the last argument, which serves as a precondition.

Delivering(*airfare*, *product*, s) statement is a fluent, meaning that the product is delivered by airfare only in situation s. Actions have preconditions identifying when these are physically possible. Equation 4.1 is an example of applying preconditions. In general, the dynamics of the domain of interest are captured by applying a set of axioms:

1. An action precondition axiom for each primitive action.
2. Successor state axioms for each fluent.

3. Unique name axioms for the primitive actions.
4. Axioms describing initial situations.
5. Domain-independent, generic axioms.

Action precondition axiom defines the condition when a particular action can take place. *Poss*(a, s) defines the situation, s, where action a is possible. Successor state axiom is a way to overcome the frame problem, where inferential aspect is present. It provides a complete way of describing how a fluent is affected as a response to the execution of actions. Successor axioms describe the value of a fluent in the next situation on the basis of its value in the current situation. Unique name axioms are consequences of the basic axioms. In $Holds(f, s_1) = Holds(f, s_2)$ situations s_1 and s_2 can be different, but may refer to the same true value for the fluent f. Generic axioms can be any one of the above-mentioned axioms, but generalized from concrete applications. These axioms are useful in applying patterns to common situations, where generic axioms can be applied and specialized with minor changes.

4.6.3 Web Standards

Modeling primitives and activities axiomatization are one aspect of knowledge engineering. In addition to this, ontology language syntax is to be defined. Taking into consideration the importance of Web environment, the syntax must be formulated using existing Web standards for information representation. Through ontology language syntax, software engineers can conceptually organize the knowledge expressed by frame-based and first-order logic. The syntax of proposed ontology language is based on XML and is a new language called SCML. The main principle on which SCML is based is the object-attribute-value (O-A-V) triplet. Objects may have other objects and/or other O-A-V triplets. Objects can be physical entities, concepts, events, and actions. Object contains attributes, which are for characterizing the object. Attribute may have one or more possible values. By assuming that terms used in O-A-V statements are based on the formally specified meaning, that is, ontologies, these triplets can be semantically processed by machine agents. Currently, ontologies applied to the World Wide Web based on O-A-V formalism are creating a new Semantic web paradigm [11].

The ontology itself is expressed as an XML schema definition (XSD) file. A candidate for SCOL is the (RDF), which has better expressiveness for representing problem taxonomy introduced earlier in this chapter and is a well-accepted Web standard.

Problem taxonomy, depicted in Fig. 4.4, is the classification of ontologies from highly generic SCOR levels to more specific levels represented by IDEF formalisms. Problem taxonomy is not discussed in this chapter and is a topic for future research as part of ontology language specification. SCML will be utilized for developing ontologies as well as for implementing the problem taxonomy. The latter will serve as a meta-model for building the ontology server—a Web-enabled knowledge portal, an environment for capturing, assembling, storing, and utilizing ontologies.

4.6.4 Specification in Ontology Language—Terminology

Artificial intelligence [26] defines ontology as the study of the kinds that exist. This definition defines the static nature of the domain that ontology captures. Frame-based and description logic can cope with this task. In addition to this, we advocate the capability of ontologies to capture the dynamics of the SC domain, for which first-order logic is utilized. Chandra and Tumanyan [6] define these two aspects of SC to be modeled as organization and problem ontologies, respectively. The implementation of these two aspects with SCML syntax is described in this section.

4.6.4.1 Organization Ontology

Organization ontology is designed to describe the structure of SC domain and its subdomains. Frame logic and object-oriented approach is utilized for describing the structure in terms of classes and properties. Organization ontologies focus on a minimal terminological structure, often just a taxonomy [13]. Chandra and Tumanyan [5] propose SC system taxonomy for capturing concepts and their subsumption relationships. System taxonomy is a classification hierarchy of SC domain characteristics and problems. It provides a syntactically homogeneous view of SC.

The terminology for organization ontology as identified in Chandra and Tumanyan [5] is as follows:

1. UML class notations denoting classes and their relationships.
2. Names of classes.
3. Names of attributes describing classes.

4.6.4.2 Problem Ontology

Problem ontologies allow adding axioms and rules to convert taxonomies into epistemological constructs, which are studies of kinds of knowledge that are required for solving problems in the world, and discovering something, or an idea embedded in a program (heuristic). Problem ontology is captured by means of situation calculus. The terminology is mostly adopted from Pinto and Reiter [29]. The vocabulary of statements and predicates is left open for future extension in case the existing terminology fails to represent SC domain specifics. Standard statements are *Do*(a, s), *Consume*(p, s), *Release*(r, s), *Produce*(p, r, s), *Start*(a, s), etc. Standard predicates are *Poss*(a, s), *Occurs*(a, s), *Actual*(s) *Holds*(f, t), *During*(t, s), *Enables*(a, s), etc.

Extended terminology can be added to express specific activities that are involved in SC. The sequence of these statements and predicates can be used to present new actions, such as sequences of letters make words, thus trying to conceptualize any activities that may happen in SC.

4.6.5 Supply Chain Markup Language

For ontology representation, different programming languages and standards have been utilized, Ontolingua and KIF [10] and OIL [11]. XML is emerging as a standard for communication between heterogeneous systems and is widely used on the Internet. These environments present new opportunities for knowledge representation and acquisition. This opportunity has three aspects. First, XML has enough expressiveness to represent knowledge captured by UML and situation calculus. Second, XML documents can easily be translated into knowledge representation format and parsed by problem-solving environments or domains. Third, XML can directly connect with data storage repositories (RDBMS or ERP systems), thus enabling database queries to be more expressive, accurate, and powerful. These three objectives can be achieved by enhancing the semantic expressiveness of XML, especially XSD. This chapter proposes a new SCML for presenting knowledge about SC. The specification of SCML is formulated as an XSD depicted in Fig. 4.6.

SCML is the computational implementation of the system taxonomy introduced by Chandra and Tumanyan [5]. Seven components (Input, Output, Process, Function, Environment, Agent, and Mechanism) of system adopted from Nadler [*28*] constitute the data model. Axiom entity is utilized for specifying axioms and algorithms.

A fragment of SCML is depicted in Fig. 4.7. It defines the entity "Axioms," elements it may have, and entities it may contain. Axioms entity class may have one or many "Rules" ("unbounded") entities, which may have "Attributes" entities (zero or many). "Argument" entity may have two attributes: "Name" and "Description." The entity "Rule" may have one and only one "Body" entity, and two attributes.

4.7 Case Study: Automotive Industry Supply Chain

A prototype of the proposed approach is developed with a case study describing a decision support system (DSS) for steel processing and shipment (SPS) SC. SCOR definitions are used for building high-level process flow. Process models are designed for activity view [19]. IDEF graphical models are utilized for decomposing SCOR processes into tasks presenting low-level flows of activities. UML class diagrams are used in transforming concepts and their relationships regarding SPS "schedule production" problem into software constructs. The UML class diagram utilization is twofold. It can be used as a pseudo-code for developing a software application. It can also serve as a semantic network for the scheduling production

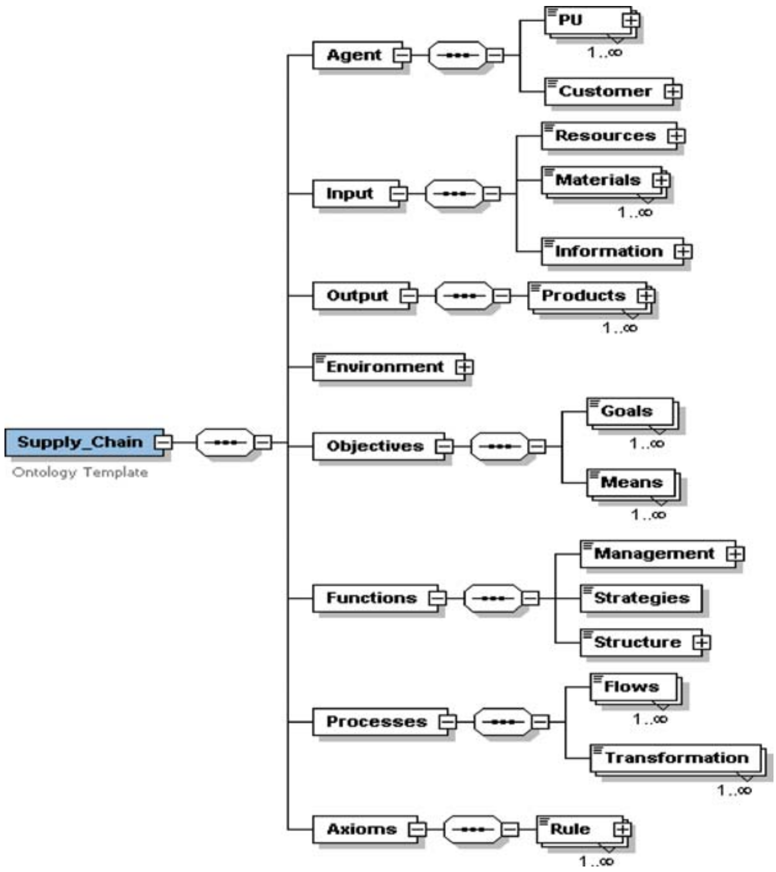


Fig. 4.6 Data schema for supply chain markup language

domain, and be a part of the domain ontology. For the task ontology example, two specifications are demonstrated, situation calculus and SCML.

4.7.1 Problem Statement

The SPS is a multistage manufacturing process. The automobile SC consists of several raw material suppliers and a stamping plant (Fig. 4.8), which bus steel from it to produce assembly components.

The stamping plant consists of blanking, pressing, and assembly departments. The blanking department cuts the raw steel into rectangular pieces. The pressing

```xml
<xs:element name="Axioms">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Rule" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Body"/>
                        <xs:element name="Attribute" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="Name" type="xs:string" use="optional"/>
                                <xs:attribute name="description" type="xs:string" use="optional"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="Number" type="xs:string" use="optional"/>
                    <xs:attribute name="Name" type="xs:string" use="optional"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" use="optional"/>
        <xs:attribute name="ID" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
```

Fig. 4.7 Supply chain markup language (SCML) fragment: axioms

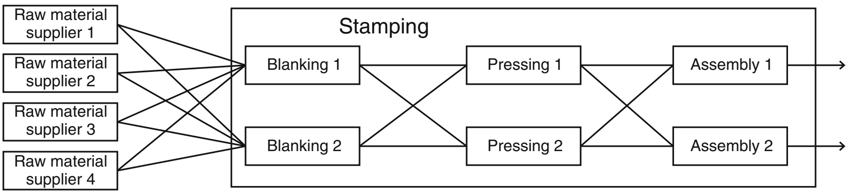


Fig. 4.8 Steel processing and shipment supply chain

department stamps the blanks into parts. Welding and other operations are performed on stamped parts at the metal assembly department.

4.7.2 Supply Chain Model

The case study described is concerned with specific problems in SC. These are supplier selection, raw materials delivery, production of steel subassembly components, and delivery using various carriers. The SPS-SC model is a projection of the SCOR process taxonomy regarding these specific issues identified in the problem statement. Fig. 4.9 depicts the sequence in which these issues are addressed in the SC.

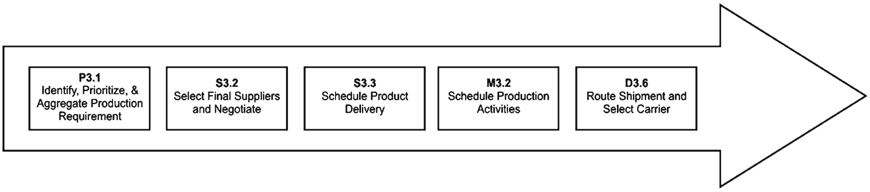


Fig. 4.9 Supply chain operations reference-model (SCOR) for steel shipment and processing supply chain

P3.1 is a process of identifying, prioritizing, and considering as a whole with constituent parts, all sources of demand in the creation of the steel product. S3.2 is the identification of the final supplier(s) based on the evaluation of supplier qualifications and the generation of a contract defining costs and terms and conditions of product availability. S3.3 is scheduling and managing the execution of the individual deliveries of product against the contract. The requirements for product deliveries are determined on the basis of the detailed sourcing plan. M3.2 are plans for the production of specific parts, products, or formulations in specified quantities and planned availability of required sourced products, and the scheduling of operations to be performed in accordance with these plans. Scheduling includes sequencing, and, depending on the factory layout, any standards for setup and run. In general, intermediate production activities are coordinated prior to the scheduling of operations to be performed in producing a finished product. D3.6 is the process of consolidating and routing shipments by mode and location. Carriers are selected and shipments are routed.

These processes are captured from SCOR level three representations. In SCOR specification, ontology designers can find performance attributes and best practices for each process. Ontology at this level is considered as the highest-level abstraction of knowledge capturing. This chapter discusses only low-level ontologies that can be ultimately used by decision modeling agents.

4.7.2.1 M3.2—Schedule Production Activity Process Model

Process model development consists of two stages: (1) two IDEF0 diagrams are developed for decomposing the "Schedule Production Activity" process into tasks and activities and (2) UML class development for representing the process model in a suitable format for building information system elements. This chapter advocates the pivotal role of ontology in information system design. Consequently, these process models will be used for designing domain and task ontologies. The IDEF0 diagram in Fig. 4.10 depicts a simple decomposition of the "Schedule Production Activity" process decomposed into the following:

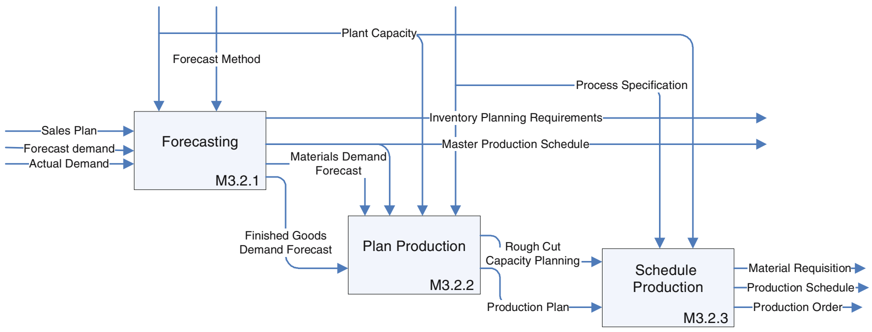


Fig. 4.10 Integrated definition0 (IDEF0) diagram for the "Schedule Production Activity" process

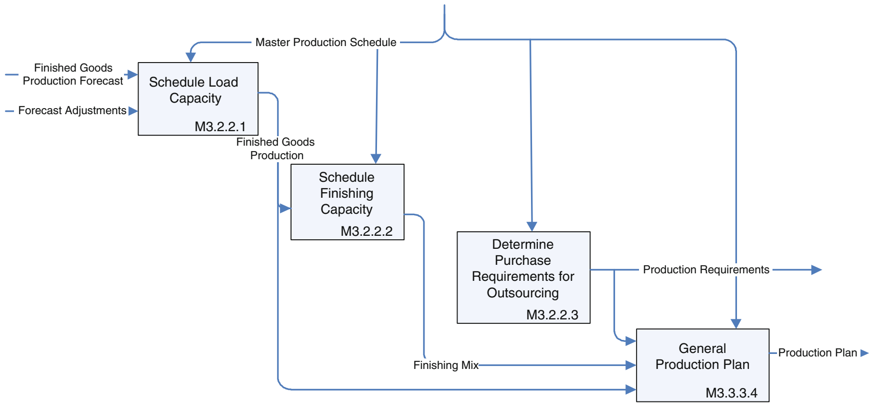


Fig. 4.11 Integrated definition0 (IDEF0) diagram for the "plan production" process

1. Forecast finished goods production and material requirements.
2. Plan production.
3. Schedule production.

Relationships among these activities identify input and output information necessary for accomplishing these tasks. Further decomposition of one of its tasks is depicted in Fig. 4.11. Four activities are identified in the "Plan Production" task: schedule load capacity, schedule finishing capacity, determine outsourcing requirements, and generate production plan.

The second part of workflow modeling is the representation of business processes with constructs that can be used for designing ontologies as information system components. The UML model development starts with studying the process models and their information needs. A thorough analysis of SPS-SC model reveals a list of parameters necessary for building the hierarchy of concepts of scheduling production (M3.2.3) process. Concepts are gathered into classes, and classes are

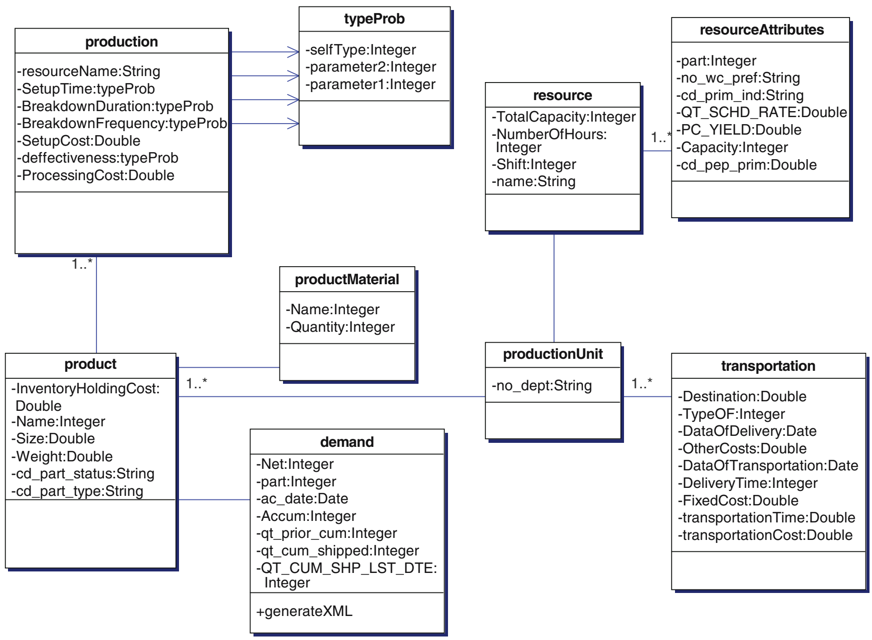


Fig. 4.12 Unified modeling language (UML) class diagram for schedule production process (M3.2.3)

related to each other. An analyzed system for the steel SC problem is depicted in Fig. 4.12. The central class is the production unit, which has transportation facilities. Association link defines a type of relationship, where transportation is presented with one parameter inside productionUnit class. This parameter is an object which has multiple attributes. Resource class is associated with productionUnit class, and has object parameter resourceAttribute. Product class is associated with production Unit with many-to-one relationship. Product has object parameters demand, product Materials, and production. TtypeProb class is for defining the distribution function of three attributes for production class, namely, BreakdownDuration, Breakdown Frequency, and defectiveness.

4.7.3 Supply Chain Ontology Engineering

SC ontology engineering consists of the development of its three components: semantic network, which is concepts and their relationships; axioms defining constrains on concepts and other horizontal relationships among them; and algorithms, which are step-by-step procedures for managing tasks and activities. A UML model defines the first component. Formalizing axioms and algorithms can be implemented through thorough analysis of developed process models.

The proposed framework describes steps necessary for formalizing ontologies. Scenario narration is provided in the problem statement section. SPS-SC informal knowledge representation is accomplished by examining process models. Observations held on scheduling production problem domain are written down as English sentences. Examples of informal representation of knowledge are as follows:

- For every product there should be a demand.
- If a product is ordered and its resource is busy with other product without order, switch the resource load.
- Inventory level should be less than the maximum allowed.
- Resource utilization cannot be more than its capacity.
- If a product is assigned to a resource, all materials should be available.
- Processes can start when resources and materials are available.

Axioms and algorithm capture is a process of a search of rules held in the domain of interest for which ontology is to be built. Rules and regulations defined in axioms and algorithms are based on concepts identified in the UML model and are formulated in the form of equations relating these concepts to each other.

The theory for axiom and algorithm representation is based on situation calculus and predicate calculus for representing the dynamically changing world [22]. Situation theory views domain as having a state (or situation). When the state is changed, there is necessity for an action. Predicate theory defines conditions on which specific actions can be taken. According to the framework introduced in this chapter, ontology is to capture both dynamics.

Examples of formal representation of axioms for the above-described situation are provided below.

$$Exist(demand, Product)$$

$$Less(MaxInventory, CurrInventory)$$

The first axiom states that if there is a product, its demand should exist. The second axiom constrains current inventory with the maximum inventory level. Axioms are reusable knowledge constructs and can be used for various problem representations, and so have to be shared among these problems.

An example of an algorithm can be the formula according to which order size is calculated. Inventory replenishment algorithm assumes checking the inventory level periodically. If it is less than a predefined level, place an order equal to a specified value. This narrated knowledge can be formalized using ontology calculus as follows:

$$Poss\{\ do[(L \times AVG + z \times STD) = s] > Il\} \equiv MakeOrder(s - Il)$$

where s is the reorder level, L is lead time, AVG, STD are forecasted demand means and standard deviation, and z is customer service indicator. If inventory level (IL) is less than the calculated reorder level, an order is placed (Order), which is equal to the difference of reorder and inventory levels.

```
<SupplyChain>
    <Axioms>
        <Rule Number="1" Name="Service level is 100%">
            <Argument Name="Inv" Description="Inventory"/>
            <Argument Name="Dm" Description="Demand"/>
            <Body>Inv&gt;=Dm</Body>
        </Rule>
        <Rule Number="2" Name="Product Resouce assignment">
            <Argument Name="P" Description="Product"/>
            <Argument Name="R" Description="Resource"/>
            <Body>FOR EACH P EXIST R</Body>
        </Rule>
        <Rule Number="4" Name="Process Production Unit correspondance/">
            <Argument Name="R" Description="Resource"/>
            <Argument Name="PU" Description="Production Unit"/>
            <Body>FOR EACH R EXIST PU</Body>
        </Rule>
        <Rule Number="5" Name="Resource utilization should be less then its capacity">
            <Argument Name="R_U" Description="Resource utilization"/>
            <Argument Name="R_C" Description="Resource capacity"/>
            <Body>R_U&gt;=R_C</Body>
        </Rule>
        <Rule Number="6" Name="Processes can start when resource and materials are available">
            <Argument Name="R" Description="Resource"/>
            <Argument Name="M" Description="Material"/>
            <Argument Name="Pr" Description="Process"/>
            <Body>For each Pr exist all R, M</Body>
        </Rule>
        <Rule Number="10" Name="If a product is assigned to a resource all materials should be available">
            <Argument Name="P" Description="Product"/>
            <Argument Name="R" Description="Resource"/>
            <Argument Name="M" Description="Material"/>
            <Body> IF P.resource CHECK ALL P.materials</Body>
        </Rule>
        <Rule Number="17" Name="Each product should have demand, internal or external">
            <Argument Name="P" Description="Product"/>
            <Body>FOR EACH P EXIST (P.demand or P.order)</Body>
        </Rule>
    </Axioms>
</SupplyChain>
```

Fig. 4.13 Ontology fragment: axioms

Ontology calculus, like IDEF process model, can document a process or a task. For using these processes (or tasks) in information system and applying process management techniques, a software construct is required for their representation.

Informally represented and presented by ontology calculus, rules are introduced as SCML data files in Fig. 4.13. This XML file structure and content is defined by SCML schema depicted in Fig 4.7.

Axioms are building blocks for ontology engineering. Along with concepts and relations, axioms can present the knowledge in a formalism that can be accessed and processed by software agents.

4.8 Conclusion

As a result of this research effort, an approach is proposed for modeling SC as a collection of ontology constructs. The process of SC workflow analysis and knowledge models design is presented. It consists of (1) modeling SC as a set of processes (SCOR), (2) modeling SC business processes from the generic process to specific tasks and activities (IDEF), (3) transforming process models into software metamodels (UML), and (4) engineering ontologies based on previous analysis.

The research described in this chapter is an organic part of overall research initiative of SC DSS design. Ontologies developed on the basis of principles presented in this chapter have been utilized for managing SPS-SC at an industrial sector. This research complements the creation of ontology development environment, where domain experts, knowledge workers, and software engineers will be able to work collectively on building enterprise ontology.

References

1. Basu A, Kumar A (2002) Research Commentary: Workflow Management Issues in e-Business. Information Systems Research 13(1): 1–14.
2. Carlsen S (1997) Conceptual Modeling and the Composition of Flexible Workflow Models, Doctoral dissertation.
3. Casati F, Ceri S, Pernici B, Pozzi, G (1996) Deriving active rules for workflow enactment. In: Proceedings of 7th International Conference on Database and Expert Systems Applications, pp. 94–110 (Springer-Verlag, New York, NY).
4. Chandra C, Kamrani AK (2003) Knowledge Management for Consumer-Focused Product Design. Journal of Intelligent Manufacturing 14: 557–580.
5. Chandra C, Tumanyan A (2003) System taxonomy development and application. In: Proceedings of the 12th Annual Industrial Engineering Research Conference IERC-2003 (Portland, OR).
6. Chandra C, Tumanyan A (2004) Ontology driven knowledge design and development for supply chain management. In: Proceedings of the 13th Annual Industrial Engineering Research Conference IERC-2004 (Houston, TX).
7. Crubézy M, Musen MA (2003) Ontologies in Support of Problem Solving. SMI Report Number: SMI-2003-0957.
8. Dussart A, Aubert BA, Patry M (2004) An Evaluation of Inter-Organizational Workflow Modelling Formalisms. Journal of Database Management 15(2): 74–104.
9. Erdmann M, Studer R (1999) Ontologies as conceptual models for XML documents. In: Proceedings of the KAW '99 12th Workshop on Knowledge Acquisition, Modelling and Management (Banff, Canada).
10. Farquhar A, Fikes R, Rice J (1997) The Ontolingua Server: A Tool for Collaborative Ontology Construction. International Journal of Human-Computer Studies 46(6): 707–727.
11. Fensel D, van Harmelen F, Horrocks I, McGuiness DL, Patel-Schneider PF (2001) OIL: An Ontology Infrastructure for Semantic Web, IEEE Intelligent Systems 16(2): 38–45.
12. Fox M, Barbuceanu M, Teigen R (2000) Agent-Oriented Supply-Chain Management. The International Journal of Flexible Manufacturing Systems 12: 165–188.
13. Gangemi A, Guarion N, Masolo C, Oltramari A (2003) Sweetening WorldNet With Dolce. Al Magazine, 24(3): 13–24.
14. Gruber TR (1993) A Translation Approach to Portable Ontologies. Knowledge Acquisition 5(2): 199–220.
15. Grüninger M, Fox MS (1995) Methodology for the design and evaluation of ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95 (Montreal, Canada).
16. Grüninger M, Fox MS (1996) The Logic of Enterprise Modelling, In: *Modelling and Methodologies for Enterprise Integration*. P. Bernus and L. Nemes (eds.), Cornwall, Great Britain, Chapman and Hall.
17. Grüninger M, Atefi K, Fox MS (2000) Ontologies to Support Process Integration in Enterprise Engineering. Computational & Mathematical Organization Theory 6: 381–394.
18. Guarino N (1998) Formal ontology and information systems, Proceedings of FOIS'98, Amsterdam (IOS Press).

19. IBM (2002) Business Modeling with UML: A Business Process Centered Architecture, Technical report.
20. Kang I, Park Y, Kim Y (2003) A Framework for Designing a Workflow-Based Knowledge Map. Business Process Management 9(3): 281–294.
21. Kim Y, Kang S, Kim D (2000) WW-FLOW: Web-Based Workflow Management Runtime Encapsulation. IEEE Internet Computing 4(3): 55–64.
22. Lesperance Y, Levesque HJ, Lin F, Scherl RB (1995) Ability and Knowing How in the Situation Calculus. Studia Logica 66(1): 165–186.
23. Levesque HJ, Reiter R, Lesperance Y, Lin F, Scherl RB (1994) GOLOG: A Logic Programming Language for Dynamic Domains. Journal of Logic Programming 31(1–3): 59–84.
24. Lin F, Yang M, Pai Y (2002) A Generic Structure for Business Process Modeling. Business Process Management Journal 8(1): 19–41.
25. Mayer RJ, Benjamin PC, Caraway BE, Painter MK (1995) A Framework and a Suite of Methods for Business Process Reengineering. In: *Business Process Reengineering: A Managerial Perspective*. B. Kettinger and V. Grover (eds.), Idea Publishing Group, Harrisburg, PA, pp. 245–290.
26. McCarthy J, Patrick JH (1969) Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: *Machine Intelligence*. B. Meltzer and D. Michie (eds.), Edinburgh University Press, pp. 463–502.
27. Meersman R (2002) *Semantic ontology tools in information system design*, STAR Lab Technical Report, Vrije Universiteit Brussel, Belgium, Brussels.
28. Nadler G (1970) *Work Design: A System Concept*, Richard D. Irwin Inc., Homewood, IL.
29. Pinto J, Reiter R (1993) Temporal reasoning in logic programming: A case for the situation calculus. In: Proceedings of the Tenth International Conference on Logic Programming, David S. Warren (ed.), pp. 203–221 (The MIT Press).
30. Presley AR, Liles DH (2001) A holon-based process modeling methodology, International Journal of Operations and Production Management 21(6): 565–581.
31. Shunk DL, Kim J, Nam HY (2003) The application of an integrated enterprise modeling methodology—FIDO—to supply chain integration modeling. Computers & Industrial Engineering 45: 167–193.
32. Tan JC, Harker PT (1999) Designing Workflow Coordination: Centralized Versus Market-Based Mechanisms. Information Systems Research, 10(4): 328–342.
33. Uschold M, Grüninger M (1996) ONTOLOGIES: Principles, Methods and Applications. Knowledge Engineering Review 11(2): 93–155.
34. Wand Y, Weber R (2002) Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda. Information Systems Research 13(4): 363–376.
35. Zha XF (2002) A Knowledge Intensive Multi-Agent Framework for Cooperative/Collaborative Design Modeling and Decision Support of Assemblies. Knowledge-Based Systems 15(8): 493–506.

Chapter 5
Data-Mining Process Overview

Ali K. Kamrani[1] and Ricardo Gonzalez[2]

Abstract This chapter gives a description of data mining and its methodology. First, the definition of data mining along with the purposes and growing needs for such a technology are presented. A six-step methodology for data mining is then presented and discussed. The goals and methods of this process are then explained, coupled with a presentation of a number of techniques that are making the data-mining process faster and more reliable. These techniques include the use of neural networks and genetic algorithms, which are presented and explained as a way to overcome several complexity problems that the data-mining process possesses. A deep survey of the literature is done to show the various purposes and achievements that these techniques have brought to the study of data mining.

5.1 Introduction

During the last few years, data mining has received more and more attention from different fields, especially from the business community. This commercial interest has grown mainly because of the awareness of companies that the vast amounts of data collected from customers and their behaviors contain valuable information. If this information can be somehow made explicit, it will be available to improve various business processes.

Data mining deals with the discovery of hidden knowledge, unexpected patterns, and new rules from large databases. It is regarded as the key element of a much more elaborate process called knowledge discovery in databases, or KDD, which is closely linked to data warehousing. According to Adriaans and Zantinge [1], data mining can bring significant gains to organizations, for example, through better-targeted marketing and enhanced internal performance. They stated in their book

[1] Associate Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004

[2] Department of Industrial and Manufacturing Systems Engineering, University of Michigan, Dearborn, Michigan 48128

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

that the long-term goal of data mining is to create a self-learning organization that makes optimal use of the information it generates.

Recent publications on data mining concentrate on the construction and application of algorithms to extract knowledge from data. Skarmeta et al. [18] developed a data-mining algorithm for text categorization. Andrade and Bork [2] used a data-mining algorithm to extract valuable information on molecular biology from large amounts of literature. Lin et al. [14] developed an efficient data-mining algorithm to measure proximity relationship measures between clusters of data. Delesie and Croes [3] presented a data-mining approach to exploit a health insurance database to evaluate the performance of doctors in cardiovascular surgeries nationwide.

The emphasis given by most authors and researchers on data mining focuses on the analysis phase of data mining. When a company uses data mining, it is important to also see that there are other activities involved in the process. These activities are usually more time-consuming and have an important influence on the success of the data-mining procedure.

This chapter is organized as follows: The following section introduces the data-mining concept as well as outlines the advantages and disadvantages of its use in the knowledge-extraction process from databases. This section also introduces some basic expertise requirements that any company should possess in order to use data mining effectively. The next section discusses the different stages involved in the data-mining process. Some data-mining techniques and methods used during the mining phase of the process are then discussed. Finally, some conclusions are presented to emphasize the importance of the techniques and methods presented in previous sections.

5.2 Data Mining

In the past, data mining has been referred to as knowledge management or knowledge engineering. Until recently, it has been an obscure and exotic technology, discussed more by theoreticians in the artificial intelligence fields. Fayyad et al. [4] defined data mining as a step in the KDD process consisting of applying computational techniques that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns or models over the data. Adriaans and Zantinge [1] gave a more general definition used by many researchers. They stated that data mining is the process of searching through details of data for unknown patterns or trends. They stressed the importance of having an efficient method of searching in large amounts of data until a sequence of patterns emerges, whether complete or only within an allowable probability.

Many times, large databases are searched for relationships, trends, and patterns, which prior to the search are neither known to exist nor visible. These relationships or trends are usually assumed to be there by engineers and marketers, but need to be proven by the data itself. The new information or knowledge allows the user community to be better at what it does. Often, a problem that arises is that large databases are searched for very few facts that will give the desired information. Moreover, the algorithm and search criteria used in a single database may change when a new trend or pattern is to be studied. Also, each database may need a different search criterion as well as new algorithms that can adapt to the conditions and problems of the new data.

More often than not, humans find it difficult to understand and visualize large data sets. Furthermore, as Fayyad and Stolorz [5] described, data can grow in two dimensions defined as the number of fields and the number of cases for each one of these fields. As they explained, human analysis and visualization abilities do not scale to high dimensions and massive volumes of data.

A second factor that is making data mining a necessity is the fact that the rate of growth of data sets completely exceeds the rates that traditional "manual" analysis techniques can cope with. This means that if a company uses a regular technique for extracting knowledge from a database, vast amounts of data will be left unsearched, as the data growth surpasses the traditional mining procedures. These factors call for a need of a technology that will enable humans to tackle a problem using large amounts of data without disregarding or losing valuable information that may help solve any kind of problem involving large data sets.

Yevich [20] stated that "data mining is asking a processing engine to show answers to questions we do not know how to ask." He explained that instead of asking in normal query language a direct question about a single occurrence on a database, the purpose of data mining is to find similar patterns that will somehow answer the desired questions proposed by the engineers or marketers. If the questions or the relationships asked to be found on a database are too specific, the process will be harder and will take more time. Moreover, a lot of important relationships will be missed or disregarded.

The interest on data mining has risen in the past few years. During the 1980s, many organizations built infrastructural databases, containing data about their products, clients, and competitors. These databases were a potential gold mine, containing terabytes of data with much "hidden" information that was difficult to understand. With the great strides shown by artificial intelligence researchers, machine-learning techniques have grown rapidly. Neural networks, genetic algorithms, and other applicable learning techniques are making the extraction of knowledge from large databases easier and more productive than ever.

Data mining is being used widely in the USA, while in Europe, it has been used to a less extent. Large organizations such as American Express and AT&T are utilizing KDD to analyze their client files. In the UK, the BBC has applied data-mining techniques to analyze viewing figures. However, it has been seen that the use of KDD brings a lot of problems. As much as 80% of KDD is about preparing data and the remaining 20% is about mining. Part of these 80% is the topic that will be analyzed and discussed in the next section.

It is very difficult to introduce data mining into a whole organization. A lot of data-mining projects are disregarded as options because of the following additional problems [1, 13]:

- Lack of long-term vision: company needs to ask themselves "what do we want to get from our files in the future?"
- Struggle between departments: some departments do not want to give up their data.
- Not all files are up-to-date: data is missing or incorrect; files vary greatly in quality.
- Legal and privacy restrictions: some data cannot be used for reasons of privacy.
- Poor cooperation from the electric data processing department.
- Files are hard to connect for technical reasons: there is a discrepancy between a hierarchical and a relation database, or data models are not up-to-date.
- Timing problems: files can be compiled centrally, but with a 6-month delay.
- Interpretation problems: connections are found in the database, but no one knows their meaning or what they can be used for.

In addition to these common problems, the company needs to have a minimum level of expertise on the data-mining processes. Scenarios in which the area expert does not have a specific question and asks the analyst to come up with some interesting results are sentenced to fail. The same holds true for situations where the expert provides the data analyst with a set of data and a question, expecting the analyst to return the exact answer to that question. According to Feelders et al. [6], data mining requires knowledge of the processes behind the data, in order to

- determine useful questions for analysis;
- select potentially relevant data to answer these questions;
- help with the construction of useful features from the raw data; and
- interpret results of the analysis, suggesting possible courses of action.

Knowing what to ask and what to expect from the information in a database is not enough. Knowledge of the available data from within is also required or at least desired. This will enable the expert and the data analyst to know where the data is and have it readily available depending on the problem being studied.

Finally, data analysis expertise is also desired. Hand [9] discussed that phenomena such as population drift and selection bias should be taken into account when analyzing a database. Data-mining expertise is required in order to select the appropriate algorithm for the data-mining problem and the questions being raised.

5.3 Data-Mining Methodology

Data mining is an iterative process. As the process progresses, new knowledge and new hypothesis should be generated to adjust to the quality and content of the data. This means that the quality of the data being studied will determine the time and precision of any given data-mining algorithm; and if the algorithm is flexible enough, important information about a problem will be found even if the central question is not fully answered.

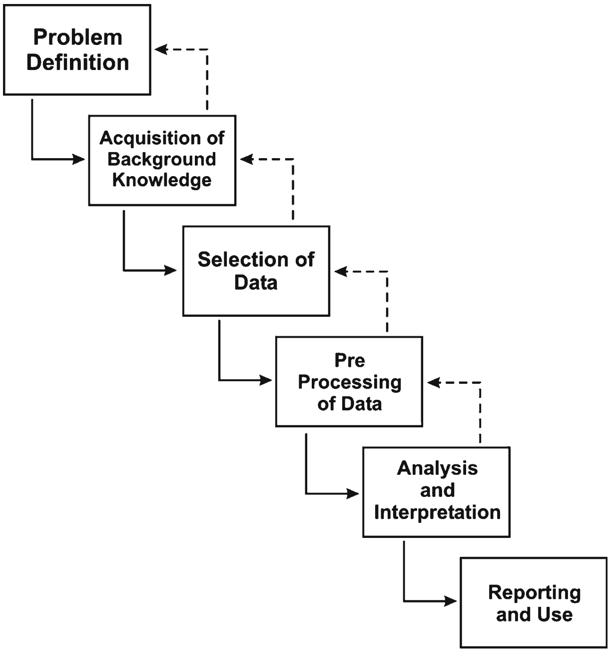


Fig. 5.1 Major steps in the data-mining process

Fayyad et al. [4] developed a structured methodology outlining the different steps and stages of the data-mining process. They outlined a six-step methodology that involved defining the problem to be solved, the acquisition of background knowledge regarding this problem, the selection of useful data, the preprocessing of the data, the analysis and interpretation of the results, and the actual use of these results. In their methodology, they stressed the importance of the constant "jump-backs" between stages. Feelders et al. [6] then used this methodology to explain each step of the data-mining process. Figure 5.1 shows that the mining stage of the process (Analysis and Interpretation) is just one of the basic stages of the data-mining methodology. The discussion on the stages of this methodology will show that all phases play a major role in the process, especially those that come before the mining stage.

5.4 Problem Definition

The problem definition phase is the first stage of the data-mining process. During this stage, the objectives of using data mining on the desired problem are identified. These are questions or assumptions that, as it was mentioned earlier, are known to exist by marketers and engineers, but that need to be proven by the data.

Most of the time, the initial question asked in a data-mining project should be very vague. This will help the data-mining process because it will disregard large amounts of data that are not useful to the problem. This way, the selection of data and the preprocessing of data will work with a set of data that has been initially "dissected" to help solve the initial problem.

During the problem definition stage, it is also important to know how the results of the data-mining process are going to be used. Glymour et al. [8] discussed the different common uses of the results of a data-mining process. Some of these include the following:

- Intervention and prediction: Results can lead to an intervention of the system being studied. Also, they can predict certain behaviors of the system.
- Description and insight: Results give an intelligent description and insight about the topic being studied.

Glymour et al. [8] also stressed the fact that one should be cautious about the source of the data. Data may be bias, an issue that will directly affect the results of the data-mining process. Biased descriptions and insights will lead to biased and possibly harmful predictions about a system. Another issue that may arise is the problem of causality. Feelders et al. [6] suggested a closer look at the data before assuming the results given by the data-mining process. This will ensure that the conclusions drawn by the process are not just the result of chance.

The problem definition stage sets the standards and expectations of the data-mining process. To an extent, this stage helps the users know the quality of the data being studied. If many iterations are required, and the problem definition ends up being too vague without getting acceptable results, the problem may lie on the quality of the data and not in the definition of the problem.

5.5 Acquisition of Background Knowledge

As it was mentioned in the previous stage of the data-mining process, possible bias and selection effects of the data being studied should be known. This knowledge will give the development team the possible limitations of the data under consideration.

Another important type of knowledge that is important to have before any selection of data is the typical causal relations found on data. Heckerman [10] proposed Bayesian Networks as a solution to this problem. They allow the incorporation of prior knowledge, which may signal possible causality found on a given result from the data. The acquisition of prior knowledge will also prevent the occurrence of "knowledge rediscovery," in essence, the data-mining algorithm will tackle a problem with a number of assertions that will prevent it from having to relearn certain patterns that are known to exist. Feelders et al. [6] proposed a method called rule induction, in which "the user would have to guide the analysis in order to take causal relations into account."

The acquisition of knowledge plays a critical role in the data-mining process. This stage can help directly on the time it takes the process to give positive results. It also prevents the process to learn facts and rules that are already known to be true.

5.6 Selection of Data

After the background knowledge is known, the data-mining process reaches the important stage of selecting the data that will be used and analyzed to give an answer to the problem under consideration.

This selection of the relevant data should be "open-minded" because the purpose of data mining is not for the human to solve the problem, but rather to let the data speak for itself. With the background knowledge in place, the data-mining process will prevent the human expert from introducing new unaccounted biases that could harm the conclusions made by the process.

Subramanian [19] and Yevich [20] proposed the use of a data warehouse as an ideal aid for selecting potential relevant data. However, a data warehouse is rarely available for use in the present time, so companies have to go through the process of "creating" one before the selection of data to achieve an acceptable selection of data. If a data warehouse is not readily available at the selection stage, the process will be a long one, and the data-mining process will suffer a big delay.

The selection of data is a crucial step in the data-mining process. Assuming the previous steps are performed properly, data selection narrows down the range of the potential conclusions to be made in the following steps. It also sets the range where these conclusions may be applicable.

5.7 Preprocessing of Data

Even when a data warehouse that has all the relevant data of the problem is available, it is often required to preprocess the data before in can be analyzed.

This stage also allows the expert the freedom of adding new attributes to the process, which will, in a way, help the data-mining procedure. These additions constitute certain relations between data that may be difficult for the data-mining algorithm to assert. Some algorithms, such as the classification tree algorithm, fail in the assertion of certain relations that may be important in the data-mining process [6]. Other algorithms take long amounts of time to make the assertions, so if the knowledge is readily available by the expert, it is better to add it directly as an attribute rather than letting the algorithm make the assertion.

Much of the preprocessing in the data-mining process is because many of the relations between entities in a database are one-to-many. Data-mining algorithms often require that all data concerning one instance of the entity should be stored in one record, so that the analysis can be done in one big table that contains all the records regarding the possible instances of a single entity.

In order for the preprocessing to be successful, the expert should use domain knowledge and common sense to determine the possible attributes and the creation of records that will enable the data-mining process to be successful over time.

5.8 Analysis and Interpretation

The analysis phase follows a long process of problem definition, selection of data, and preprocessing of data. At this time of the process, at least 70% of the time used in data-mining process has elapsed. During this phase, it is critical for the expert to have experience and knowledge on the subject area being studied, on data analysis, and of course, on data mining.

Knowledge on the subject being studied is required primarily to interpret results, and most important, to assert which results should be taken into account for further study for possible corrective actions if they are necessary at one point. Data analysis experience is principally required to explain certain strange patterns found on the data, and to give importance to more interesting parts of the data being studied. Finally, data-mining experience and expertise are required for the technical interpretation of the results. This technical interpretation is in essence, as Feelders et al. ([6] mention in their paper, the translation of the results to the language of the domain and the data expert.

The analysis and interpretation stage of the data-mining process is where the actual mining takes place. After the data has been selected, preprocessed, and the problem to be solved is known, this stage tries to find certain patterns, similarities, and other interesting relations between the available data. All these patterns are usually translated into rules that are used in the last phase of the data-mining process.

5.9 Reporting and Use

Results of a data-mining process have a wide range of uses. Results can be used in a simple application, such as being input for a decision process, as well as in an important application, like a full integration into an end-user application.

The results of a data-mining process can also be used in a decision support system or a knowledge-based system [13, 20]. This field of application enables the learning process of the knowledge-based system to be faster and more efficient since it will not have to wait or adjust to certain biases that a human expert may have. The knowledge obtained will be in some cases more accurate, and if the background knowledge obtained during the data-mining process is reliable, then the results will have a higher reliability as well.

As was mentioned in an earlier section of this paper, the data-mining process has been used as a tool for various purposes. Its results can be used to predict patterns and behaviors, or just to organize, sort, and choose certain amounts of data to prove an assertion made by an expert in the field under consideration.

In almost any event, the results of the data-mining process should be presented in an organized fashion. This implies the use and development of a well-defined user interface. A good and robust user interface is critical in the data-mining process as well as in a wide range of problems. It enables the users to report and interact with the programs effectively, which improves the success chances of the process.

In summary, the data-mining process is a long and iterative process. As Fig. 5.1 depicts, the process goes both ways, so for example, after selecting the data to be studied, the expert can "go back" to the previous stage of the process to add more background knowledge that may have been forgotten. Also, if the definition of the problem is too specific and no results are found, the expert has the advantage of going back to redefine the problem.

The selection stage and the preprocessing stage are the most critical steps in the data-mining process. The two steps account for almost 80% of the time used in the data-mining effort. Thus, special attention should be given to these two steps, if the data-mining process is to be a success.

The last two steps of the process are usually "routine work," if the required data analysis and mining expertise are used properly. Many researchers have discussed these two steps, but the good performance and use of the analysis and the results of the process is completely dependent on the previous stages. Without a good definition of the problem or with an inadequate selection of data, the results are not going to be useful.

Companies have to realize that the data-mining process is a long and sometimes complicated process. Expertise in many fields is required for the process to be successful. Workers have to be patient, as the process may take a long time to be able to come up with useful answers. Management has to support the procedure and know that analyzing and reporting results are not the only two parts of the data-mining process. In fact, they have to understand that these two steps are just the end work of an extensive and complex process.

5.10 Data-Mining Techniques

Adriaans and Zantinge [1] summarized the various techniques used for tackling and solving complex data-mining problems, stressing that the selection of a technique should be very problem specific.

Some of the techniques available are query tools, statistical techniques, visualization, online analytical processing (OLAP), case-based learning, decision trees, neural networks, and genetic algorithms. The complexity of these techniques varies, and a good selection of the technique used is critical to arrive at good and significant results. Table 5.1 shows the techniques available for data mining and their basic advantages and limitations.

Table 5.1 Data-mining techniques

Data-mining technique	Characteristics Advantages/disadvantages
Query tools	• Used for extraction of "“shallow”" knowledge. • SQL is used to extract information.
Statistical techniques	• SPC tools are used to extract deeper knowledge from databases. • Limited but can outline basic relations between data.
Visualization	• Used to get rough feeling of the quality of the data being studied.
Online analytical processing (OLAP)	• Used for multi-dimensional problems. • Cannot acquire new knowledge. • Database cannot be updated.
Case-based learning	• Uses k-nearest neighbor algorithm. • A search technique rather than a learning algorithm. • Better suited for small problems.
Decision trees	• Good for most problem sizes. • Gives true insight into nature of the decision process. • Hard to create trees from a complex problem.
Neural networks	• Mimics human brain. • Algorithm has to be trained during the encoding phase. • Complex methodology.
Genetic algorithms	• Use Darwin's evolution theory. • Robust and reliable method for data mining. • Requires a lot of computing power to achieve anything of significance.

SQL structured query language, *SPC* statistical process control, *OLAP* online analytical processing

The simplest technique available is the use of query tools for a rough analysis of the data. Just by applying simple structured query language (SQL) to a data set, one can obtain a lot of information. Before applying more advanced techniques, there is a need to know some basic aspects and structures of the data set. This was stressed in the previous section, and it was regarded as background knowledge. SQL is a good technique for discovering knowledge that is in the "surface" of the data, which is information that is easily accessible from the data set.

Scott and Wilkins [17], Adriaans and Zantinge [1], and Yevich [20] have stated that for the most part, about 80% of the interesting information can be abstracted from a database using SQL. However, as Adriaans and Zantinge [1] stressed in their book, the remaining 20% of hidden information can be extracted only by using more advanced techniques; and for most problems, this 20% can prove of vital importance when solving a problem or extracting valuable knowledge for future use.

Statistical techniques can be a good simple start for trying to extract this important 20% of knowledge from the data set. Patterns can be found in the data being studied with the help of histograms, Pareto diagrams, scatter diagrams, check sheets, and other statistical tools. If important relations are not found using statistical process control (SPC) tools, at least some information will be learned, and some relations that are definitely not in the data will be known, so that more advanced techniques do not have to look for weak relations in the data being studied.

Visualization techniques are very useful for discovering patterns in data sets and are usually used at the beginning of a data-mining process to get a feeling of the quality of the data being studied. This technique uses SPC as well to gain knowledge from databases based on simple but important patterns.

The OLAP tools are widely used by companies. They support multidimensional problems that involve many sorts of information requested by managers and workers at the same point in time. OLAP tools store the data being studied in a special multidimensional format, and managers can ask questions from all ranges. A drawback of OLAP tools is that the data cannot be updated. Also, as Fayyad and Stolorz [5] mentioned, OLAP is not a learning tool, so no new knowledge can be created, and new solutions cannot be found. Essentially, if the data has the solutions, OLAP will work well, but if the solutions are not there, then OLAP is useless and obsolete.

Case-based learning uses the k-nearest neighbor algorithm [1] to assert relations in a database. This algorithm, however, is not really a leaning algorithm. Russell and Norvig [16] used the algorithm as a search method rather than as a learning technique. However, as they mention in their book, this search technique proves to be very useful since the data set itself is used as reference. In essence, the search technique only searches the data set space, and thus, it does not get "corrupted" by outside data or knowledge.

A problem with the search technique used in case-based learning is its complexity. The algorithm searches and compares every single record or input from the database with each other, in order to find relations between records in a data set, so as the amount of data increases, the algorithm's complexity increases as well. According to Russell and Norvig [16], this search technique leads to a quadratic complexity, which is obviously not desirable when searching large data sets. This technique has been used widely but only in small problems that have small databases as input for data assertion.

Decision trees are usually used for classification purposes. The database is classified into certain fields that will enable the expert to assert certain behaviors and patterns found in the database. In essence, the data will be divided into categories and to make an assertion or find a pattern in the data, one has to follow a path in the decision tree to arrive at a conclusion. The path taken represents the assertions, facts, and other information used to make the desired conclusion.

Many algorithms have been proposed for the creation of such decision trees. Adriaans and Zantinge [1] used the tree induction algorithm to create trees used by car companies to predict customers' behaviors. An advantage of this approach is the complexity of most of the algorithms. Most decision tree algorithms are very effective, and have an n Log n complexity.

An important requirement for a decision tree to be successful is to have good knowledge of the problem at hand and of the data available. A lot of assertions and decisions regarding relations between data have to be made by the expert, and the technique relies heavily on these decisions.

Genetic algorithms follow the theory of evolution proposed by Darwin. His theory is based on the "natural selection" process of evolution, which essentially states that each species has an overproduction of individuals and in a tough struggle for life, only those individuals that are best adapted to the environment survive. The same principle can and has been adapted by many researchers that have used genetic algorithms as a learning tool on various data-mining problems.

Holmes et al. [11] developed and used a genetic algorithm to search and learn certain patterns for epidemic surveillance. By searching large databases, the algorithm creates a number of rules regarding possible causes, risks, and solutions for certain problems. Koonce et al. [12] used genetic algorithms in data mining for learning manufacturing systems. Michalski [15] developed the Learnable Evolution Model (LEM), a genetic approach that differs in many aspects to the Darwinian model used by most researchers. The LEM uses machine learning to improve the evolutionary process. In data mining, this "new" approach improves the performance and the ability of the algorithm to learn and find interesting patterns.

Vila et al. [22] used genetic algorithms and neural networks to improve the quality of the data used in data mining, as they stressed that finding patterns and solutions on incomplete and/or unreliable data will result in useless conclusions. Fu [7] compared a greedy search method with a genetic-based approach for two-dimensional problems involving large data sets. He explained that the improvements shown by the genetic algorithm approach are vast. They include a more robust methodology, a faster method for finding patterns, and more reliable results. Yuanhui et al. ([21] combined a neural network with a genetic algorithm to mine classification rules. They showed that the genetic approach generates better rules than do the decision tree approach.

The advantages and disadvantages of genetic algorithms follow those of natural selection in general. An important drawback is the large overproduction of individuals. Genetic algorithms work with populations of chromosomes, and large amounts of data are used to solve even the easiest problems. Another problem with genetic algorithms is the random character of the searching process. As Adriaans and Zantinge [1] discuss, the end-user of a genetic algorithm does not really see how the algorithm is creating and selecting individuals for finding patterns in large amounts of data.

An advantage of genetic algorithms is their reliability. One can be sure that if a solution exists in a large database, the genetic algorithm will find it. This is of course, assuming that all the requirements of a genetic algorithm have been addressed and used properly. Another important advantage concerning reliability and robustness is that genetic algorithms do not need to have previous "experience" on the problem at hand. This means that a genetic algorithm used for data mining will eventually find a pattern (if there is any) on the data even if the problem at hand is brand new and no past solutions have been found.

As databases expand, researchers have grown more and more dependent in artificial intelligence to solve these complicated data-mining problems. Artificial intelligence techniques potentially have low complexities, and more importantly, they resemble the human way of thinking, which may make these techniques the most reliable of all, at least from the standpoint of humans.

5.11 Conclusions

This chapter presented an introduction on data mining, the data-mining methodology, and the goals and techniques available for solving all kinds of data-mining problems. Why data mining is important and needed as a tool for improving the way business is being done is discussed first. This importance lies in the fact that in order to be successful, a company has to be aware of what the customers are thinking and saying about their product. This may seem like a simple task, but many times companies misunderstand customer responses, and at that time, data mining can present a better look at the available data.

As companies grow, in both time and size, the data collected by the company grows at an incredibly fast rate. Human understanding of this data can only go to a point, at which point data mining becomes a necessity.

The data-mining process is an iterative one. As was presented in a previous section of this chapter, constant "jump-backs" can be made between stages of the process, until accurate results are found or until the database is shown to be unreliable. Also, it has to be noted that the most important stages of the process are the selection and the preprocessing of the data, instead of the mining stage itself.

Data mining can serve various purposes. It can be used for association, classification, clustering, and summarization, among other goals. However, the means of getting these results can vary from problem to problem. Query language, SPC, visualization, and OLAP are the techniques used to achieve a greater understanding on the data being studied, but as problems grow larger in size and as data becomes more complex, new approaches have to be used. In essence, the computerization has to come back to what humans do best, that is, analyzing small portions of data using many resources that regular techniques do not possess, and that such techniques as neural networks and generic algorithms do.

References

1. Adriaans, P. and D. Zantinge, *Data Mining*, Harlow: Addison-Wesley, 1996.
2. Andrade, M. and P. Bork, "Automated extraction of information in molecular biology," FEBS Letters, 476: 12–17, 2000.
3. Delesie, L. and L. Croes, "Operations research and knowledge discovery: A data mining method applied to health care management," International Transactions in Operational Research, 7: 159–170, 2000.
4. Fayyad, U., D. Madigan, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," AI Magazine, 17: 37–54, 1996.
5. Fayyad, U. and P. Stolorz, "Data mining and KDD: Promise and challenges," Future Generation Computer Systems, 13: 99–115, 1997.
6. Feelders, A., H. Daniels, and M. Holsheimer, "Methodological and practical aspects of data mining," Information & Management, 37: 271–281, 2000.
7. Fu, Z., "Dimensionality optimization by heuristic greedy learning vs. genetic algorithms in knowledge discovery and data mining," Intelligent Data Analysis, 3: 211–225, 1999.

8. Glymour, C., D. Madigan, D. Pregibon, and P. Smyth, “Statistical themes and lessons for data mining,” Data Mining and Knowledge Discovery, 1: 11–28, 1997.
9. Hand, D.J., “Data mining: Statistics and more?” The American Statistician, 52: 112–118, 1998.
10. Heckerman, D., *Bayesian Networks for Knowledge Discovery, Advances in Knowledge Discovery and Data Mining*, Menlo Park, CA: AAAI Press, pp. 273–305, 1996.
11. Holmes, J.H., D.R. Durbin, and F.K. Winston, “The learning classifier system: An evolutionary computation approach to knowledge discovery in epidemiologic surveillance,” Artificial Intelligence in Medicine, 19: 53–74, 2000.
12. Koonce, D.A., C. Fang, and S. Tsai, “A data mining tool for learning from manufacturing systems,” Computers & Industrial Engineering, 33: 27–30, 1997.
13. Kusiak, A., *Computational Intelligence in Design and Manufacturing*. Wiley-Interscience Publications, pp. 498–526, 1999.
14. Lin, X., X. Zhou, and C. Liu, “Efficient computation of a proximity matching in spatial databases,” Data & Knowledge Engineering, 33: 85–102, 2000.
15. Michalski, R.S., “Learnable evolution model: Evolutionary processes guided by machine learning,” Machine Learning, 38: 9–40, 2000.
16. Russell, S., P. Norvig, *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall, 1995.
17. Scott, P.D. and E. Wilkins, “Evaluating data mining procedures: Techniques for generation artificial data sets,” Information and Software Technology, 41: 579–587, 1999.
18. Skarmeta, A., A. Bensaid, and N. Tazi, “Data mining for text categorization with semi-supervised agglomerative hierarchical clustering,” International Journal of Intelligent Systems, 15: 633–646, 2000.
19. Subramanian, A., L.D. Smith, A.C. Nelson, J.F. Campbell, and D.A. Bird, “Strategic planning for data warehousing,” Information and Management, 33: 99–113, 1997.
20. Yevich, R., “Data Mining,” in *Data Warehouse: Practical Advice from the Experts*, Prentice Hall, pp. 309–321, 1997.
21. Yuanhui, Z., L. Yuchang, and S. Chunyi, “Mining classification rules in multi-strategy learning approach,” Intelligent Data Analysis, 2: 165–185, 1998.
22. Vila, M.A., J.C. Cubero, J.M. Medina, and O. Pons, “Soft computing: A new perspective for some data mining problems,” Vistas in Astronomy, 41: 379–386, 1997.

Chapter 6
Intelligent Design and Manufacturing

Emad S. Abouel Nasr[1] and Ali K. Kamrani[2]

Abstract The progressive opportunity of international markets has led to significant new competitive pressures on industry. Recently, this has seen changes in organizational structures at the product design level through the prologue of computer-integrated manufacturing (CIM) and concurrent engineering (CE) philosophy, and is now seeing changes in industry structures as companies build worldwide manufacturing relationships. Automatic feature recognition from computer-aided design (CAD) systems plays an important key toward CAD/computer-aided manufacturing (CAM) integration. Different CAD packages store the information related to the design in their own databases. Structures of these databases are different from each other. As a result, no common or standard structure that can be used by all CAD packages has been developed so far. For that reason, this chapter will propose an intelligent feature recognition methodology to develop a feature recognition system which has the ability to communicate with various CAD/CAM systems. The system takes a neutral file in initial graphics exchange specification (IGES) format for 3-D prismatic parts as input and translates the information in the file to manufacturing information. The boundary representation (B-rep) geometrical information of the part is analyzed by a feature recognition program on the basis of object-oriented and geometric reasoning approaches. A feature recognition algorithm is used to recognize different features such as step and holes.

6.1 Introduction

The main objective of any manufacturing organization is to produce high-quality products at the lowest possible cost. The growing complexity of achieving this objective, with sharply rising costs and increased competition, has forced the indus–try to look for alternatives to the traditional approaches to design, manufacturing, and management. Many industries are adopting a concurrent engineering (CE) approach to develop and

[1] Assistant Professor, Mechanical Engineering Department, Faculty of Engineering at Helwan, Helwan University, Helwan, Cairo, Egypt

[2] Associate Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5, © Springer Science + Business Media, LLC 2008

produce new products in the most efficient manner. Computer-aided process planning (CAPP) systems can help reduce the planning time and increase consistency and efficiency [20]. However, the main problem of transferring the computer-aided design (CAD) data to a downstream computer-aided manufacturing (CAM) system, in order to develop computer-integrated manufacturing (CIM) environment, is the lack of neutral formats as well as content to convey the CAD information [27, 39].

Recently, this has seen changes in organizational structures at the product design level through the prologue of CIM and CE philosophy, and is now seeing changes in industry structures as companies build worldwide manufacturing relationships [*36*, *38*]. Taking these issues into considerations leads to the recognition that the integration between design and manufacturing needs to be made to ensure business competitiveness. In order to achieve the integration of design and manufacturing, understanding of how the manufacturing information can be obtained directly from the CAD system must be addressed [13, 29, 31].

CAD and CAM systems are based on modeling geometric data. The usefulness of CAD/CAM systems is the ability to visualize product design, support design analysis, and link to the generation of part programmers for manufacturing [32]. However, CAD/CAM systems need the standardization that makes them have the ability to communicate to each other. Different CAD or geometric modeling packages store the information related to the design in their own databases and the structures of these databases are different from each other. As a result, no common or standard structure has so far been developed yet that can be used by all CAD packages. On the contrary, the conventional approach to feature extraction is accomplished by the human planner examining the part and recognizing the features designed into the part. Automated feature recognition can best be facilitated by CAD systems capable of generating the product geometry based on features, thereby making it possible to capture information about tolerance, surface finish, etc. [12]. However, such CAD systems are not yet mature and their wide usage in different application domains remains to be seen. For that reason, in this chapter, a methodology for feature analysis and extraction of prismatic parts for CAM applications is developed and presented. This approach aims to achieve the integration between CAD and CAM.

Including this introductory section, the chapter is organized into six sections. The problem statement is addressed in Sect. 6.2. Section 6.3 describes the literature review of the previous research efforts in the area of feature extraction and recognition. The proposed approach for extraction of manufacturing entities from initial graphics exchange specification (IGES) file as a standard format is presented in Sect. 6.4. The implementation of the suggested approach is demonstrated through an example in Sect. 6.5. Finally, Sect. 6.6 presents conclusions.

6.2 Problem Statement

Different CAD or geometric modeling packages store the information related to the design in their own databases. Structures of these databases are different from each other. As a result, no common or standard structure that can be used by all CAD packages has been developed so far. For that reason, this research will try to develop an intelligent feature recognition methodology which has the ability to communicate with the different CAD/CAM systems by using object-oriented and geometric reasoning approaches.

6.3 Literature Review

The recognized features and their relationships due to the feature recognition process are used to restructure the part. A feature has a higher level of abstraction than a geometric primitive in the traditional solid model [8]. Features not only represent the shape but should also contain information on its functions and interrelationship with other features [11].

The most widely used representations methods are boundary representation (B-rep) and constructive solid geometry (CSG). The CSG of the solid model is specified with a set of Boolean operations and a set of 3-D primitive solids as shown in Fig. 6.1. On the contrary, B-rep is one of the solid modeling methods that

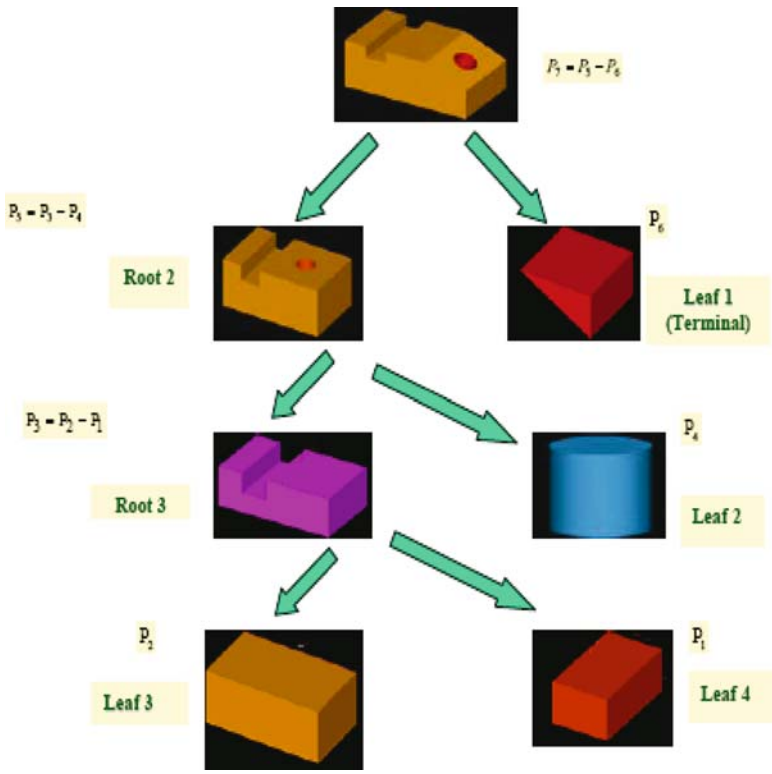


Fig. 6.1 Constructive solid geometry (CSG) representation

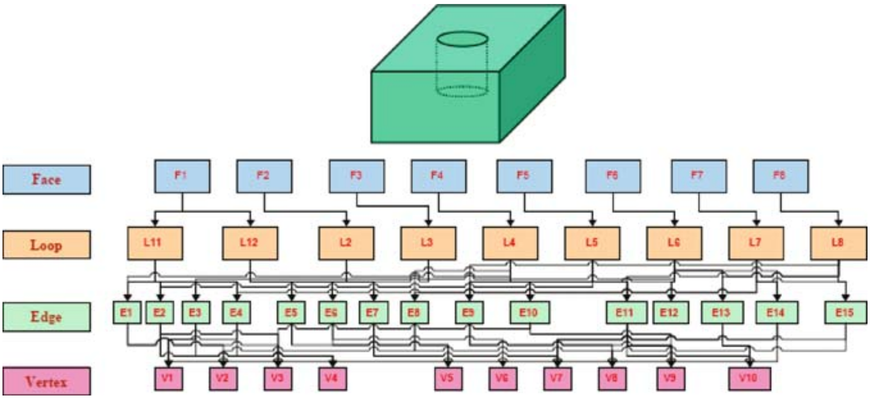


Fig. 6.2 A boundary representation (B-rep)

are broadly used to generate a solid model of a physical object [18]. The B-rep describes the geometry of an object in terms of its boundaries, which are the vertices, edges, and surfaces as shown in Fig. 6.2 [7]. In the next subsections, literature survey in the area of feature representations will be described.

6.3.1 Feature Representation by B-Rep

Tseng and Joshi [47] developed a method for feature recognition of mill-turned parts. B-rep was used to create rotational and prismatic components. This method was based on machining volume generation approach to recognize and classify features. The feature volumes were generated by sweeping boundary faces along a direction determined by the type of machining operations. In this approach, first, the part was segmented into several rotational machining zones. Next, the prismatic features were recognized on the bases of intermediate rotational shapes using volume decomposition and maximal volume sweeping and reconstruction. The classification of prismatic features was done by using the face adjacency relationships, while classification of rotational parts was performed by using profile edge patterns.

Aslan et al. [5] developed a feature extraction module for only rotational parts that are to be machined at turning centers. In this paper, the data interchange format (DXF) file was used to extract 2-D, which was represented by B-rep, for rotational parts. This extraction module was a part of an expert system called ASALUS. ASALUS was designed to manage the life cycle of rotational parts from design all the way to production by performing process planning using a generative approach and applying post-processing for two different computer numerical control (CNC) lathes. Prismatic features and the intersecting features were not involved in this module.

Nagaraj and Gurumoothy [34] described an algorithm to extract machinable volumes that need to be removed from a stock. This algorithm can handle both prismatic and cylindrical components by using the B-rep model. The machinable volumes can be used to automate process planning and NC tool path generation. The algorithm identified the cavity volumes in the part with respect to the outermost faces in the part and filled them with the appropriate primitive volume to obtain the stock from which the part can be realized.

Kayacan and Celik [19] developed a feature recognition system for process planning for prismatic parts. This system was achieved with the B-rep modeling method to give vectorial direction knowledge and adjacent relationships of surface using the Standard for the Exchange of Product Model Data (STEP) standard interface program.

6.3.2 Feature Representation by constructive solid geometry

Requicha and Chan [37] developed a scheme for representing surface features in a solid modeler based on CSG and for associating tolerances and other attributes with such features. Their approach treats tolerances as attributes of object features which are part of the object's surface or topological boundary. They developed a graph structure called the variation graph to represent these features and attributes.

Shah and Roger [42] developed an integrated system for form features, precision features, and material features. The solid representation of the form features is stored as a feature producing volume (CSG tree) and Boolean operators. An object-oriented programming approach is used to represent the feature descriptions. The feature relationship graph is created at the top level of the model, where both the adjacency and the parent–child dependency of the form features are stored.

6.3.3 Feature Recognition Techniques

There are two approaches for building the CAD/CAM interface. They are design by features and feature recognition [14, 22]. Both approaches focus on the concept of "features." Design by features or the so-called feature-based design is a way of using design features to accomplish the construction of parts' CAD models. At first scene, this may seem to obviate the need for subsequent feature recognition. However, such features, being primarily design-oriented, have to be converted into manufacturing features to build the interface between feature-based design and CAM applications [41]. On the contrary, feature recognition is one of the major research issues in the area of automated CAD/CAM interface.

Various approaches and algorithms are proposed by many researchers. These are *syntactic pattern recognition* [45], *volume decomposition* [21], *expert system and logic* [33], the *CSG-based approach* [35], *the graph-based approach* [16], and *the neural-network-based approach* [11]. Most suggested methods for feature recognition used the internal representation of features that was created by the CAD system which had its own feature structure. On the contrary, there are few researches which had used the standard format of product data. Some of them will be briefly discussed below.

Meeran and Pratt [30] used 2-D entities in a DXF file with no specific order and no relationship in their connectivity. The process of searching and sorting the entities is divided into three groups according to each of three orthographic views. The main objective of the approach was to recognize the machining features of prismatic parts that have planar or cylinder faces in terms of 2-D shapes which are located in the three orthographic views of the drawing. The first step of the approach is recognizing the isolated feature, then providing the library of patterns. The recognition is fundamentally based on the pattern matching approach. It cannot recognize prismatic parts and it has the drawback of using 2-D drawing.

Sheu [44] developed a CIM system for rotational parts. The parametric design and feature-based solid model were used to specify the manufacturing information required to the proposed system. In this system, the boundary of a solid model could be transferred directly into the line and arc profiles. The part model was created by using the cylinder, the cone, the convex arc, and the concave arc as primitives. This system had the ability to convert the wireframe part model into CSG representation by the feature recognition approach. Prismatic components and their interactions were not considered.

Ahmad and Haque [3] developed a feature recognition system for rotational components using DXF file. In this approach, the work geometric information of rotational parts is translated into manufacturing information through a DXF file. A feature recognition algorithm was used to recognize different features of the part from its DXF file, where geometric information of the part was stored after respective DXF codes. Finally, using the data extracted from DXF file, each feature of the part was recognized. The parts are symmetrical and were represented by two dimensions.

Mansour [28] developed simple software to link CAD packages and CNC machine tools. The main feature of this software was its ability to automatically generate part programs for machining sculptured surfaces. To achieve this objective, IGES files of simple or free-form surfaces were exploited. The data extracted from IGES files were used to graphically simulate the tool path due to the center and tip of a ball nose tool and a filleted cutter as well as the tip of a flat-end mill cutter. AUTOCAD and AUTOCAD development system (ADS) were used to convert trimmed surfaces to parametric spline surfaces. The package was not developed to handle other IGES entities which provide a wide range of free-form surfaces.

From the literature review, significant efforts have been focused on the development of fully automatic manufacturing feature recognition systems in the last two decades. No matter which approach is adopted, the geometric information always needs to be constructed early in order to advance the feature recognition. Table 6.1 summarizes the literature review according to the feature recognition method, representation type, standard type, dimension type, and feature type. From the system implementation point of view, the best system should be independent of the format, that is, any kind of data format can be used for the input information and the internal geometric feature representation can be constructed on the basis of the input data.

Table 6.1 Summary of literature review

Researcher	Feature recognition method	Represen– tation type		Standard type			Dimension type			Feature type	
		CSG	B-rep	DXF	IGES	STEP	2-D	2.5-D	3-D	R	D
Liu [25]	Heuristic		*		*				*		*
FU [12]	Heuristic		*		*	*			*		*
Bhandarkar [6]	Heuristic		*			*			*		*
Roy [40]	AI		*						*	*	
Chang [9]	Syntactic		*				*			*	
Munns [33]	Logic	*							*		*
Liu [24]	Graph	*							*		*
Joshi [16]	Graph		*						*		*
Ciurana [10]	Vol. Dec.	*	*						*		*
Liu [25]	Heuristic				*				*		*
Madurai [26]	Expert		*							*	
Sharma [43]	AI					*			*		*
Linardkis [23]	Expert			*			*	*			*
Stalely [46]	Syntactic		*						*	*	
Vankataraman [48]	Graph		*						*		*
Zhao [51]	Heuristic								*		*
Chang [8]	AI								*		*
Kayacan [19]	Expert		*			*			*		*
Abouel Nasr [2]	Heuristic	*	*		*				*		*
Hwang [15]	Expert		*						*		*
Kao [18]	Graph		*						*		*
Zhang [50]]	Logic		*		*		*				*

R = rotational P = Prismatic vol.Dec = Volume decomposition AI = artificial Intelligent

B-rep is the most independent representation because other data formats have several limitations and disadvantages. Therefore, this chapter will adopt the solid modeling representation. On the contrary, the CSG technique will be used as a tool to construct the designed parts, so hybrid CSG/B-rep will be used in this research.

6.4 The Proposed Methodology

In this section, the part design is introduced through CAD software and is represented as a solid model by using the CSG technique as a design tool. The solid model of the part design consists of small and different solid primitives combined together to form the required part design. The CAD software generates and provides the geometrical information of the part design in the form of an ASCII file (IGES) [4] that is used as standard format which provides the proposed methodology the ability to communicate with the various CAD/CAM systems.

The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program that is created specifically to extract the features from the geometrical information on the basis of the geometric reasoning object-oriented approaches. The feature recognition program is able to recognize these features: slots (through, blind, and round corners), pockets (through, blind, and round corners), inclined surfaces, holes (blind and through), and steps (through, blind, and round corners). These features that are mapped to process planning as an application for CAM are called manufacturing information. Figure 6.3 shows the structure of the proposed methodology.

The proposed methodology presented in this chapter consists of three main phases: (1) a data file converter, (2) an object form feature classifier, and (3) a manufacturing features classifier as shown in Fig. 6.4. The first phase converts a CAD data in IGES/B-rep format into a proposed object-oriented data structure. The second phase classifies different part geometric features obtained from the data file converter into different feature groups. The third phase maps the extracted features to process planning point of view.

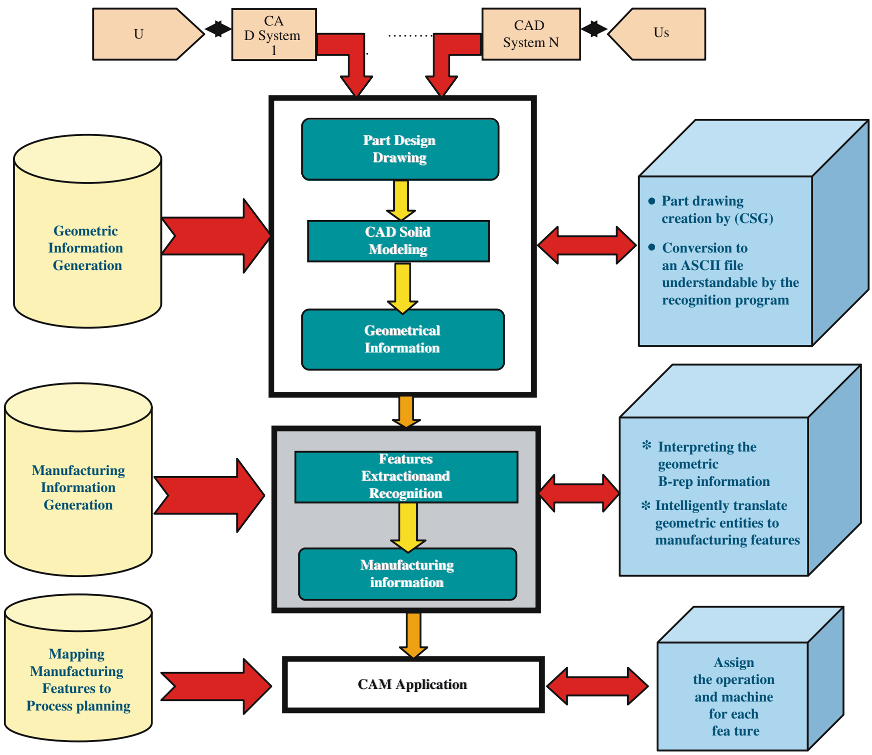


Fig. 6.3 Structure of the proposed methodology

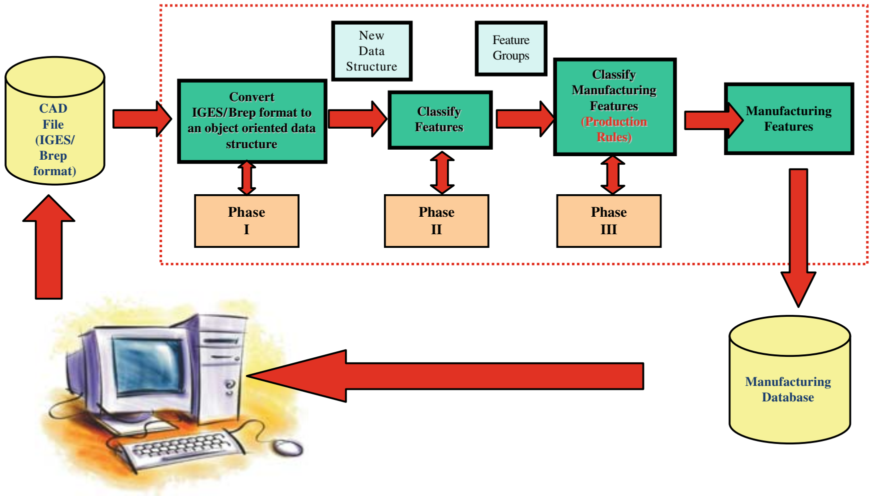


Fig. 6.4 Flowchart of extraction and classification of features

6.4.1 Conversion of Computer-Aided Design Data Files to Object-Oriented Data Structure

IGES is a standard format that can be used to define the data of the object drawing in solid modeling CAD systems in B-rep structure (IGES 5.3, 1996). Object's geometric and topological information in IGES format can be represented by the entry fields that constitute the IGES file [17]. The geometric information are in low-level entities, such as lines, planes, circles, and other geometric entities for a given object, and the topological information that defines the relationships between the object's geometric parts are represented, for example, by the loops (external loop and internal loop). An external loop provides the location of main geometric profiles and an internal loop represents a protrusion or a depression (through/blind holes) on an external loop.

6.4.2 The Overall Object-Oriented Data Structure of the Proposed Methodology

In order to have a good generic representation of the designed object for CAM applications, for example, process planning, the overall designed object description and its features need to be represented in a suitable structured database [25]. An object-oriented representation will be used in this chapter. The first step toward automatic feature extraction will be achieved by extracting the geometric and topological information from the (IGES/B-rep) CAD file and redefining it as a new object-oriented data structure as demonstrated in Fig. 6.5.

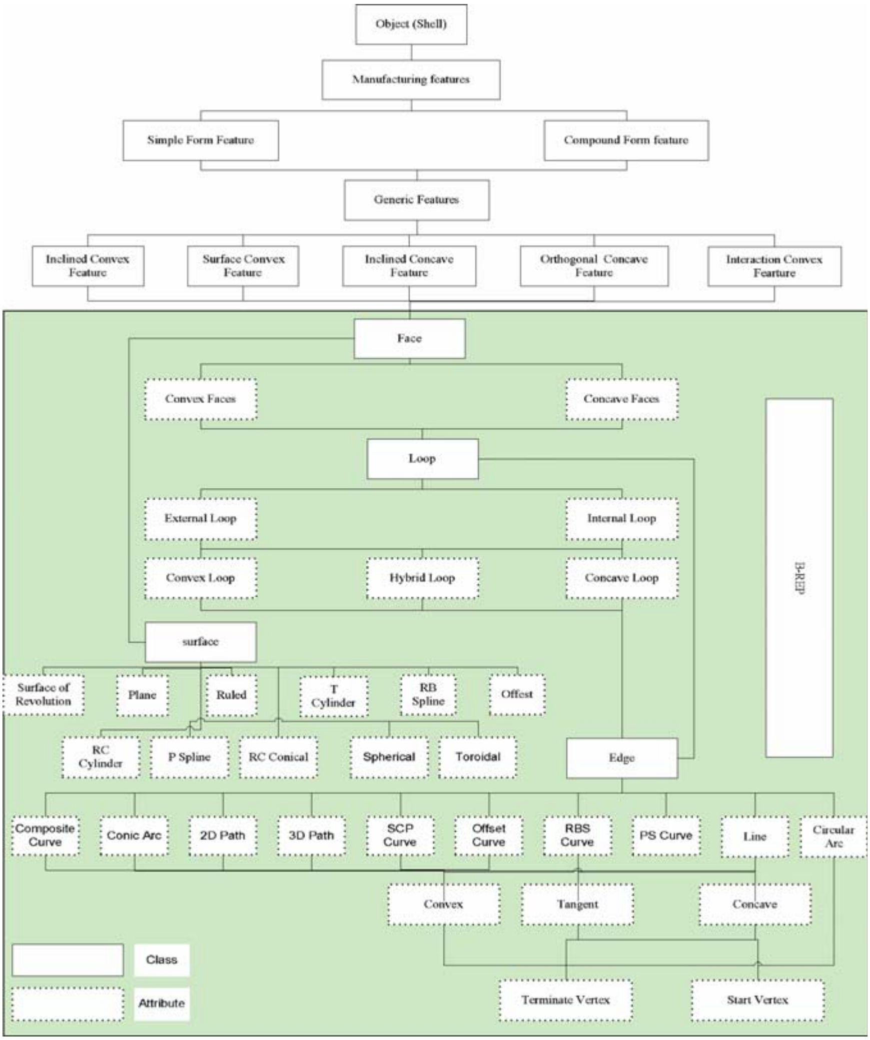


Fig. 6.5 Hierarchy of classes and attributes of the designed object

In this hierarchy, the highest level data class is the designed object (shell). An object consists of manufacturing features that can be classified into form features which decomposed of either simple or compound features. A simple feature is the result of two intersecting general geometric surfaces while compound feature is one that results from the interaction of two or more simple features (slot and pocket). Features are further classified into concave or convex as attributes in the generic feature class. Concave features consist of two or more concave faces, and convex features are decomposed of either one or more convex faces or the interaction between other features in the object. Moreover, faces of any designed part may be any type of surfaces such as plane surface, ruled surface, and p-spline surface. Also, the edge of any designed part can be any type of edges such as line, circular arc, and conic arc.

Because of the attributes of the geometric and topological entities of form features (FF), form features can be classified into two categories [12], which are interior form feature ($FF_{interior}$) and exterior form feature ($FF_{exterior}$). $FF_{interior}$ can be defined as the features which are located inside the basic face. On the contrary, $FF_{exterior}$ can be defined as the features which are formed by the entire basic surface with its adjacent faces. The basic face can be defined as the face in which there are features located in that face. Moreover, the interior form features ($FF_{interior}$) can be further classified into two low-level categories, convex interior form feature ($FF_{interior\_convex}$) and concave interior form feature ($FF_{interior\_concave}$). $FF_{interior\_convex}$ is the convex section in a basic face, whereas FFinterior_concave is the concave geometric section in the face.

The basic idea to define concave features is to identify concave faces which are defined by a concave edge that connects two adjacent faces. A concave edge is determined by the concavity test that was adapted and modified from Liu et al. [25] and Hwang [15]. Basically, the edge is defined by two vertices (start vertex and terminate vertex) expressed in 3-D solid model in terms of coordinates (X, Y, Z). On the contrary, convex features can be defined and classified as inclined, interaction, or surface. Inclined convex features are defined by a group of convex faces which are not parallel or perpendicular to the smallest surrounded envelope of the designed object. The other type of convex feature, interaction features, results from the interaction of two or more features. Surface convex features are features that locate on the exterior enclosing envelop.

6.4.2.1 Classification of Edges

Edges forms the wireframe of any 3-D solid model and any two adjacent faces can be connected by one edge. Edges can be classified as concave, convex, or tangent edges, as shown in Fig. 6.6. This classification of edges will facilitate the implementation of the proposed methodology as proposed in Fig. 6.5. To represent edge types, the normal vectors of the two faces connected to that edge and the edge direction are determined. Then by applying the connectivity test, the edge classification can be achieved. Also, the angle between the two faces (Ω) is determined by the following equation:

$$\text{Angle}(\Omega) = \cos^{-1}\left(\frac{X_1X_2 + Y_1Y_2 + Z_1Z_2}{\sqrt{X_1^2 + Y_1^2 + Z_1^2}\sqrt{X_2^2 + Y_2^2 + Z_2^2}}\right) \cdot \frac{\pi}{180} \quad (6.1)$$

where X_1, Y_1, and Z_1 are the components of the normal vector N_1 of the first face and X_2, Y_2, and Z_2 are the components of the normal vector N_2 of the second face.

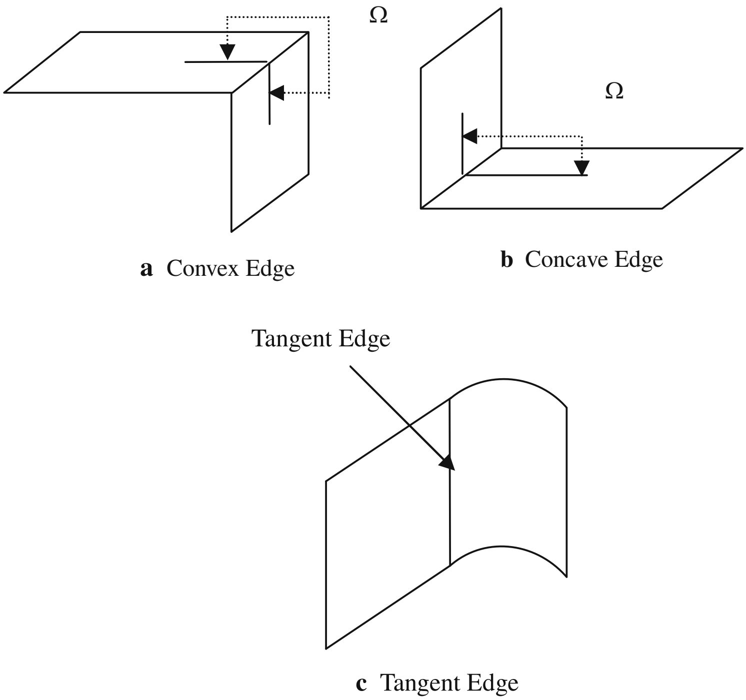


Fig. 6.6 Classifications of edges

6.4.2.2 Classification of Loops

A loop can be defined as the boundary of a face. Moreover, it can be the intersection border of the face with its adjacent faces. In this research, a loop is used as a basic indication to recognize the interior and exterior form features. The loop can be classified as proposed in this research into two categories, external loops and internal loops [12, 49]. External Loop is the exterior border of a basic face of which the loop is examined, while internal loops are located within the basic face. By examining the basic face, an external loop can be identified by the maximum margin of the basic face and the internal loop can be recognized by the interior interface boundary of the basic face with its interior features.

6.4.3 Definition of the Data Fields of the Proposed Data Structure

Generally, faces are the basic entities that constitute the features, which are further defined by edges that are represented in terms of vertices, which are defined in terms of coordinates in CAD file. Therefore, the hierarchy of the designed object that was described in the previous section (Fig. 6.4) represents multilevel of different classes. All classes, except for the superclass representing an object as a whole, are objects of classes that are higher up in the data structure. For example, each edge object is represented in terms of vertex objects. Table 6.2 displays the data attributes required for each class in the object-oriented data structure that is defined before.

Table 6.2 Definitions of classes and attributes

Class name	Attribute	Type
Point	X_ Coordinates	(Real)
	Y_ Coordinates	(Real)
	Z_ Coordinates	(Real)
Vertex	Inherits Point	Point
	Vertex _ID	(Integer)
Vertex_ List	Vertex_ Count	(Integer)
	Vertex_ List	(Vector of Vertex pointers)
Edge	Edge_ ID	(Integer)
	Edge_ Type	(Enumerated Constants)
	Start_ Vertex	(Vertex Pointer)
	Terminate_ Vertex	(Vertex Pointer)
	Concavity	(Enumerated Constants)
	Face_ Pointers [2]	(Array of Face Pointers)
	Loop_ Pointers [2]	(Array of Loop Pointers)
	Dimension	(real)
Edge_ List	Edge_ Count	(Integer)
	Edge_ List	(Vector of Edge Pointer)
Loop	Loop_ ID	(Integer)
	Loop_ Concavity	(Enumerated Constants)
	Loop_ Type (External or Internal)	(Enumerated Constants)
	Edge_ List	(An edge list of loop edges)
	Face_ Pointers	(Pointer to face class)
Surface	Surface_ Type	(Enumerated Constants)
Face	Face_ ID	Number
	Surface_ Pointer	(Pointer to the Surface)
	External_ Loop	(Loop Pointer)
	Internal_ Loop_ Count	Number
	Internal_ Loop_ List	(Vector of Loop Pointers)
Shell	Vertex_ List	(Object of Vertex_ List class)
	Edge_ List	(Object of Edge_ List class)
	Loop_ List	(Vector of Loop Pointers)
	Surface_ List	(Vector of Surface Pointers)
	Face_ List	(Vector of Face Pointers
	Name	(String)
	IGES_ File	(Object of IGES_ File Class)
Feature	Feature_ ID	(Number)
	Feature_ Type	(Enumerated constants)
	Feature_ Origin	(Vertex Pointer)
	Length	(Real)
	Width	(Real)

(continued)

Table 6.2 (continued)

Class name	Attribute	Type
	Height	(Real)
	Radius	(Real)
	Face_ List	(Vector of Face Pointers)
	Direction	(An object of Point Class)
Compound Feature	Feature_ ID	(Number)
	Feature_ Type	(Enumerated constants)
	Feature_ List	(Vector of Feature Pointers)
	Merging_ Feature	(Integer)

6.4.4 Algorithms for Extracting Geometric Entities from CAD File

The IGES file is sequentially read (on a line basis) and parsed into appropriate entry classes known as **DEntry** and **PEntry** as the most important and useful sections of the IGES are the directory section and the parameter section. **DEntry** represents an entry in the directory section, while **PEntry** represents an entry in the parameter section. The collection of directory entry classes is contained in a container class called **DSection.**

Similarly, the parameter entry classes are contained in the **PSection** class. A **Parser class** object is created using these classes to parse the information present in the entries and classify the information into different classes that are used to represent different entities of the diagram described by the IGES file. Two algorithms for extraction of data from the IGES file into a proper set of data structures were developed. The first algorithm is developed for extracting entries from directory and parameter sections of the IGES file and it will be addressed later in this chapter. On the contrary, the second algorithm is developed for extracting the basic entities of the designed part [1].

6.4.4.1 Algorithm for Extracting Entries from Directory and Parameter Sections

// Algorithm to extract the directory entries
// and the parameter section entries from the iges file.
// This process takes place during the construction of an object of IGESFile class.
// Each such object represents one IGES file.

1. Create a file descriptor IgesFile.
2. Create an empty dSection1 class (container to store dEntry objects).
3. Create an empty pSection1 class (container to store pEntry objects).
4. Open the Iges file for reading using IgesFile file descriptor // Read the file to scan and extract the directory and parameter sections.
5. While ReadLine line1 from the Igesfile

5.1 If line1 belongs to Directory section
5.1.1 If line1 is the first line of Dsection
5.1.1.1 Set dIndex to 1
5.1.2 ReadLine line2 from the Igesfile
5.1.3 Create an object dEntry1 of class DEntry
5.1.4 Set dEntry1 index using dIndex
5.1.5 Initialize dEntry1 using string Line1 + Line2
5.1.6 Add dEntry1 to dSection1 class
5.1.7 Set dIndex = dIndex + 1
5.2 If line1 belongs to Parameter Section
5.2.1 If line1 is the first line of PSection
5.2.1.1 Set pIndex to 1
5.2.2 Create an empty string Line2
5.2.3 while pEntry data incomplete
5.2.3.1 ReadLine Line3 from the Igesfile
5.2.3.2 Append Line3 to Line2
5.2.4 Create an object pEntry1 of class PEntry
5.2.5 Set pEntry1 index equal to pIndex
5.2.6 Initialize pEntry1 using string Line1 + Line2
5.2.7 Add pEntry1 to pSection1 class
5.2.8 Set pIndex = pIndex + 1
5.3 If line1 belongs to Terminate Section
5.4 exit while loop
6. End of while loop

6.4.5 Extracting Form Features from Computer-Aided Design Files

The edge direction and the face direction are the basic entities information that is used to extract both simple and compound form features from the object data structure. The edge directions in object models can be defined such that when one walks along an edge, its face is all the time on the left-hand side. When an edge is located in the external loop of a face, its direction will be in anticlockwise direction relative to the surrounding face. On the contrary, when an edge is located in the internal loop of a face, its direction will be clockwise [15].

The concave edge test used in research is based on the cross product of the normal vectors of the two faces joined by a given edge. This is done by applying vector geometry to the face and edge direction vectors. Figure 6.7 shows the symbols used in this test where the *i*th face is designated as F_i, its corresponding normal direction vector is defined as N_i in the upward direction with respect to the given face, and the *k*th edge is designated as E_k. For each edge of the designed part, the edge (E_k) is shared by two faces (F_i and F_j) where the order is right to left from the left side

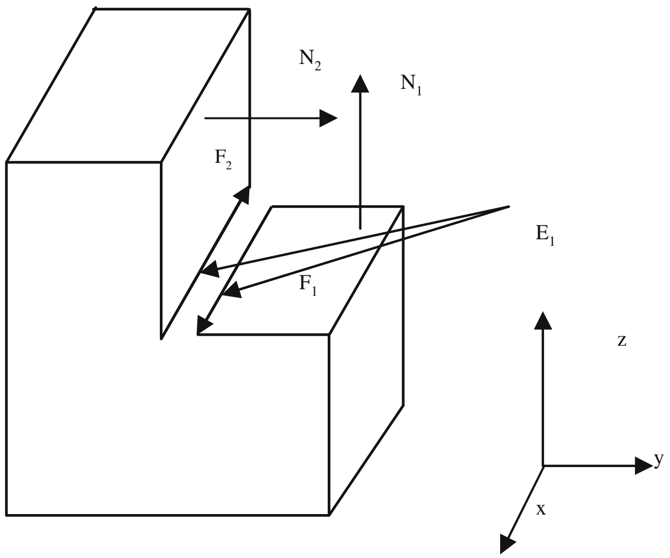


Fig. 6.7 A concave edge example

of the edge view perspective. The direction vectors of the faces are as described above (N_i and N_j). Finally, the edge's directional vector is given with respect to the face F*i* using the loop L_i that contains the edge (E_k). The following is the methodology for the concavity test:

1. The cross (vector) product (V) of the directional vectors of the faces is determined as follows:

$$V = N_i \times N_j$$

2. The direction of the edge E_k with respect to the face F_i is determined. The normal vector N*i* of face F*i* must be the first component in the cross product of step 1.
3. If the direction vector of the edge E_k from step 2 is in the same direction of cross product V, then the edge E_k is a convex edge that concludes F_i and F_j are convex faces, otherwise, it will be a concave edge and F_i and F_j are concave faces. Also, if the cross product vector V is a zero vector that means that the edge is of tangent category.

This procedure will be done on all the edges of the object to define the concave, tangent, or convex faces. Moreover, concave features will be identified by the premise that concave faces include at least one concave edge with adjacent concave faces forming a concave face set. Each concave face set defines a concave feature. Similarly, adjacent convex faces form a convex face set. Two algorithms are developed to determine the concavity of both edges and loops. The concavity edge algorithm will be discussed later in this chapter while the concavity loop algorithm can be found in Abouel Nasr and Kamrani [2].

6.4.5.1 Algorithm for Determination the Concavity of the Edge

The following algorithm is used to find the concavity of a line entity. This algorithm is used by the Line class to find the entities.

1. $\text{Length of Line} = \sqrt{[(\text{startX-termX})^2 + (\text{startY-termY})^2 (\text{startZ-termZ})^2]}$
2. concavity = UNKNOWN
3. If face1 surface type = = PLANE and face2 surface type = = PLANE
   3.1. Assign crossDir = cross product of face1 normal vector and face2 normal vector
   3.2. If crossDir = = 0
      3.2.1 concavity = TANGENT
   3.3 Else
      3.3.1 Calculate the direction vector edgeDir for the line with respect to the loop
         3.3.1.1 If crossDir is in the same direction as edgeDir
         3.3.1.1.1 concavity = CONVEX
         3.3.1.2 Else
         3.3.1.2.1 concavity = CONCAVE
4. If face1 surface type = = RCCSURFACE and face2 surface type = = PLANE
   4.1 Find dir1 (direction) that is orthogonal to the plane containing the edge and the axis of face1
   4.2 If dir1 and normal of face2 are orthogonal to each other
      4.2.1 concavity = TANGENT

6.4.5.2 Algorithms for Feature Extraction (Production Rules)

The proposed methodology is able to extract many manufacturing features. Each feature has its own algorithm (production rule). The following are the two algorithms used for extraction of both step through and step through with round corners:

Feature: **STEP THROUGH** (Fig. 6.8)

1. For every concave edge of type Line in the edge list.
2. If the two common faces (face1 and face2) of the edge (e1) are **plane** and **orthogonal** to each other
   2.1. if outer loop concave edge count equals 1 in both the faces
      2.1.1. STEP THROUGH found
      2.1.2. Create a new StepT object and add to feature list
3. End For

Feature **STEP THROUGH ROUND CORNER** (Fig. 6.9)

1. For every 3 tangent edges of type Line in the edge list (e_1, e_2, e_3)
2. If the common face of the 2 edges is quarter cylindrical surface (F_2, F_3).
   2.1 The other two faces (F_1, F_4) connected to edges are perpendicular to each other.
      2.1.1. If concave edge count of the outer loops of the four faces equals 0 each.

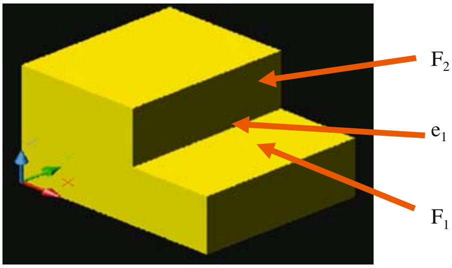


Fig. 6.8 Step through

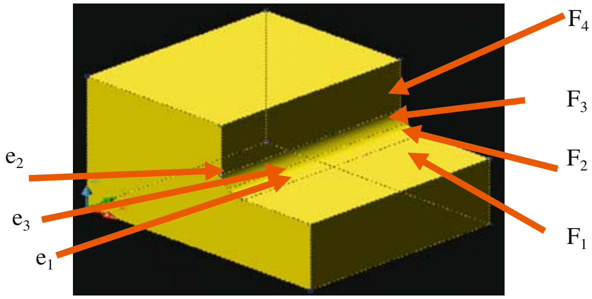


Fig. 6.9 Step through round corner

2.1.1.1. STEP THROUGH ROUND CORNER found
2.1.1.2. Create a new StepT_RC object and add to feature list.

3. End For

6.5 Illustrative Example

The proposed methodology is used for the component illustrated in Fig. 6.10. Mechanical Desktop 6 Power Pack is the CAD system used that supports B-rep and IGES translator. However, other similar CAD systems that support IGES translator can be used. The proposed methodology is developed by using Microsoft Visual C++ 6 windows based on PC environment. The designed object consists of eight different features and prismatic raw material.

Mechanical Desktop CAD system is one of the recent CAD softwares that can be used for design applications. This CAD system supports both B-rep and IGES translator (version 5.3). In Mechanical Desktop, the designed part can be represented by 2-D or 3-D drawings. In this chapter, the part designs are represented by 3-D solid models using the CSG technique as a design tool. After creating the 3-D solid model of the designed part, the CAD user has to export the 3-D solid model

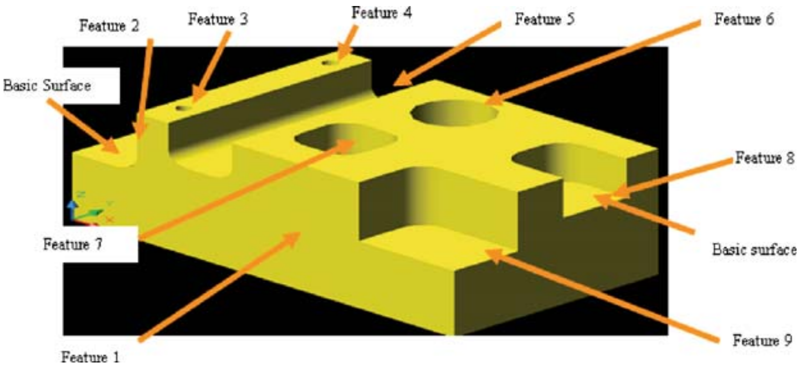


Fig. 6.10 An illustrative example

file into the IGES format provided that B-rep option version 5.3 should be highlighted. Then the user has to save the IGES file in order to be used as an input for the developed program. This file is designated as geometrical attributes file. By applying the proposed methodology, the final results are shown in Table 6.3.

6.6 Conclusion

In this chapter, a new methodology for extraction manufacturing features from IGES file format was introduced. The proposed methodology was developed for 3-D prismatic parts that were created by using solid modeling package by using the CSG technique as a drawing tool. The system takes a neutral file in IGES format as input and translates the information in the file to manufacturing information. The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program that was created specifically to extract the features from the geometrical information on the basis of the geometric reasoning and object-oriented structure approaches.

The methodology discussed has several advantages over other methods suggested in the literature. First, by using the object-oriented approach, the proposed methodology has the ability to provide a good and generic representation of the simple and compound product data in which the feature, geometry, topology, and manufacturing data are associated. Second, the proposed methodology is flexible to the variations of the IGES file format from different vendors that offer different CAD systems. Third, the proposed methodology separated the extraction module of the IGES entities of the designed part from the subsequent modules of the program. This makes the proposed methodology easily adaptable to any other standard format such as STEP or DXF. Fourth, the proposed methodology is using the IGES format as a standard input. Most feature recognition approaches found in the

Table 6.3 Manufacturing features

Feature ID	Feature type	Faces ID	Edges ID	Location	Feature name	Dimension L	W	H	R
[1]	Prismatic	[38][50][48][39][42][49]	[120][125][137]	[84] = (0,0,0)	Raw material	60	32	32	
[2]	$FF_{exterior}$	[41][2][40][42]	[7][5][126]	[6] = (9.12,0,17.88)	Step_Through (Round Corner)	32	12	7	3
[3]	$FF_{interior}$	[39][48]	[42][53]	[26] = (12.5,28,0)	Hole_Through			32	1
[4]	$FF_{interior}$	[39][48]	[43][54]	[26] = (12.5,5,0)	Hole_Through			32	1
[5]	$FF_{exterior}$	[36][33][4][34][3][35][37]	[103][13][15][11][9][108]	[13] = (15.88,0,22.88)	Slot_Through (Round Corner)	32	7	4	3
[6]	$FF_{interior}$	[32][17]	[33][17]	[26] = (35,25,17)	Hole_Blind			32	1
[7]	$FF_{interior}$	[19][10][18][21][9][20][23][8][22][25][7][24]	[63][39][37][60][35][33][66][31][29][71][27][25]	[57] = (28,8,0)	Pocket_Through (Round Corner)	32	4	4	3
[8]	$FF_{interior}$	[30][29][28][5][27][26][6][31]	[85][17][19][80][21][23][87][84][20][82][79][24][90]	[61] = (50,20,22)	Slot_Blind (Round Corner)	7	4	16.49	3
[9]	$FF_{interior}$	[44][43][1][46][45]	[132][1][3][135][131]	[2] = (45.88,9.12,17)	Step_Blind (Round Corner)	15	12	7	3

literature have been developed by the internal data structure of the CAD system. Therefore, these approaches are specific domain for these CAD systems. The proposed methodology can also extract simple curved features like round corners. Most researchers who extracted curved features tried to approximate the curvature of the surfaces which result in inaccurate representations or extraction of the manufacturing features. Finally, the proposed methodology operates in 3-D solid modeling environment which gives it a powerful ability to be used by the current manufacturing technology.

References

1. Abouel Nasr, E., & Kamrani, A., A Feature Interaction Approach for CAM Applications, 37th International Conference on Computers and Industrial Engineering, Alexandria, Egypt (2007).
2. Abouel Nasr, E., & Kamrani, A., A New Methodology for Extracting Manufacturing Features from CAD System, International Journal of Computers and Industrial Engineering, 15(1), 389–415 (2006).
3. Ahmad, N., & Haque, A., Manufacturing Feature Recognition of Parts Using DXF Files, Fourth International Conference on Mechanical Engineering, Dhaka, Bangladesh (1), 111–115 (2001).
4. ANS US PRO/IPO-100, Initial Graphics Exchange Specifications: IGES 5.3 (1996).
5. Aslan, E., Seker, U., & Alpdemir, N., Data Extraction from CAD Model for Rotational Parts to be Machined at Turning Centers, Turkish Journal of Engineering and Environmental Science, 23(5), 339–347 (1999).
6. Bhandarkar, M.P., Downie, B., Hardwick, M., & Nagi, R., Migrating from IGES to STEP: One to One Translation of IGES Drawing to STEP Drafting Data, Computers in Industry, 41(3), 261–277 (2000).
7. Chang, T.C., Expert Process Planning for Manufacturing, Addison-Wesley Publishing Company, Reading, MA (1990).
8. Chang, H.C., Lu, W.F., & Liu, F.X., Machining Process Planning of Prismatic Parts Using Case-Based Reasoning and Past Process Knowledge, Applied Artificial Intelligence, 16(4), 303–331 (2002).
9. Chang, P., & Chang, C., An Integrated Artificial Intelligent Computer Aided Process Planning System, International Journal of Computer Integrated Manufacturing, 13(6), 483–497 (2000).
10. Ciurana, J., Romeu, M.L.G., & Castro, R., Optimizing Process Planning using Groups of Precedence Between Operations Based on Machined Volume, Engineering Computations, 20(1/2), 67–81 (2003).
11. Devireddy, C.R., & Ghosh, K., Feature-Based Modeling and Neural Networks-Based CAPP for Integrated Manufacturing, International Journal of Computer Integrated Manufacturing, 12(1), 61–74 (1999).
12. Fu, M.W., Lu, W.F., Ong, S.K., Lee, I.B.H., & Nee, A.Y.C., An Approach to Identify Design and Manufacturing Features from a Data Exchanged Part Model, Computer Aided Design, 35(11), 979–993 (2003).
13. Groover, M.P., Automation, Production Systems, and Computer-Integrated Manufacturing, Prentice-Hall, Inc., New Jersey (2001).
14. Harun, W.A., & Case, K., Feature-Based Representation for Manufacturing Planning, International Journal of Production Research, 38(17), 4285–4300 (2000).
15. Hwang, J., Rule-Based Feature Recognition: Concepts, Primitives and implementation. MS Thesis, Arizona State University (1988).

16. Joshi, S., & Chang, T.C., Graph-Based Heuristics for Recognition of Machined Features from 3D Solid Model, Computer Aided Design, 20(2), 58–66 (1988).
17. Kahrs, M., The Heart of IGES, Software-Practice and Experience, 25(8), 935–946 (1995).
18. Kao, C.-Y., Kumara, S.R.T., & Kasturi, R., Extraction of 3D Object Features from CAD Boundary Representation using Super Relation Graph Method, IEEE Transaction on Pattern Analysis and Machine Intelligence, 17(12), 1228–1233 (1995).
19. Kayacan, M.C., & Celik, S.A., Process Planning System for Prismatic Parts, Integrated Manufacturing Systems, 14(2), 75–86 (2003).
20. Lee, K., Principles of CAD/Cam/CAE Systems, Addison-Wesley Publishing Co., Inc., Boston, MA (1999).
21. Lin, A.C., & Lin, S.-Y., Volume Decomposition Approach to Process Planning for Prismatic Parts with Depression and Protrusion Design Features, International Journal of Computer Integrated Manufacturing, 11(6), 548–563 (1998).
22. Lin, A.C., Lin, S.-Y., & Cheng, S.-B., Extraction of Manufacturing Features from a Feature-Based Design Model, International Journal of Production Research, 35(12), 3249–3288 (1997).
23 Linardakis, S., & Mileham, A.R., Manufacturing Feature Identification for Prismatic Components from CAD DXF Files, Advances in Manufacturing Technology VII, Proceedings of 9th National Conference on Manufacturing Research, 37–41 (1993).
24. Liu, C.-H., Perng, D.-B., & Chen, Z., Automatic Form Feature Recognition and 3D Part Recognition from 2D CAD Data, Computer and Industrial Engineering, 26(4), 689–707 (1994).
25. Liu, S., Gonzalez, M., & Chen, J., Development of an Automatic Part Feature Extraction and Classification System Taking CAD Data as Input, Computers in Industry, 29(3), 137–150 (1996).
26. Madurai, S.S., & Lin, L., Rule-Based Automatic Part Feature Extraction and Recognition from CAD Data, Computers and Industrial Engineering, 22(1), 49–62 (1992).
27. Ma, X., Zhang, G., Liu, S., & Wang, X., Measuring Information Integration Model for CAD/CMM, Chinese Journal of Mechanical Engineering, 16(1), 59–61 (2003).
28. Mansour, S., Automatic Generation of Part Programs for Milling Sculptured Surfaces, Journal of Materials Processing Technology, 127(1), 31–39 (2002).
29. Marri, A., & Kobu, B., Implementation of Computer-Integrated Manufacturing in Small and Medium Enterprises, Industrial and Commercial Training, 35(4), 151–157 (2003).
30. Meeran, S., & Pratt, M.J., Automatic Feature Recognition from 2D Drawings, Computer Aided Design, 25(1), 7–17 (1993).
31. Meeran, S., Taib, J.M., & Afzal, M.T., Recognizing Features from Engineering Drawings Without Using Hidden Lines: A Framework to Link Feature Recognition and Inspection Systems, International Journal of Production Research, 41(3), 465–495 (2003).
32. Miao, H.K., Sridharan, N., & Shah, J., CAD/CAM Integration Using Machining Features, International Journal of Computer Integrated Manufacturing, 15(4), 296–318 (2002).
33. Munns, A., Li, Y., & Wang, X.C., A Rule-Based Feature Extraction from CSG Representations and an Application in Construction, Proceedings of SPIE—The International Society of Optical Engineering, 2620(1), 269–276 (1995).
34. Nagaraj, H.S., & Gurumoorthy, B., Machinable Volume Extraction for Automatic Process Planning, IIE Transactions, 34(4), 393–410 (2002).
35. Natekar, D., Zhang, X., & Subbarayan, G., Constructive Solid Analysis: A Hierarchal, Geometry-Based Meshless Analysis Procedure for Integrated Design and Analysis, Computer Aided Design, 36(5), 473–486 (2004).
36. Reiter, W.F., Collaborative Engineering in the Digital Enterprise, International Journal of Computer Integrated Manufacturing, 16(7–8), 586–589 (2003).
37. Requicha, A.A.G., & Chan, S.C., Representation of Geometric Features, Tolerances, and Attributes in Solid Modelers Based on Constructive Geometry, IEEE Journal of Robotics and Automation, RA-2(3), 156–166 (1986).
38. Rouibah, K., & Casekey, K.R., Change Management in Concurrent Engineering from a Parameter Perspective, Computers in Industry, 50(1), 15–34 (2003).

39. Roucoules, L., Salomons, O., & Paris, H., Process Planning as an Integration of Knowledge in the Detailed Design Phase, International Journal of Computer Integrated Manufacturing, 16(1), 25–37 (2003).
40. Roy, U., & Liu, C.R., Feature-Based Representational Scheme of a Solid Modeler for Providing Dimension and Tolerancing Information, Robotics & Computer-Integrated Manufacturing, 4(3/4), 335–345 (1998).
41. Rozenfeld, H., & Kerry, H.T., Automated Process Planning for Parametric Parts, International Journal of Production Research, 37(17), 3981–3993 (1999).
42. Shah, J.J. & Roger, M.T., Expert Form Feature Modeling Shell, Computer Aided Design, 20(9), 515–524 (1988).
43. Sharma, R., & Gao, J.X., Implementation of STEP Application Protocol 224 in an Automated Manufacturing Planning System, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 216(1), 1277–1289 (2002).
44. Sheu, J.J., A Computer Integrated Manufacturing System for Rotational Parts, International Journal of Computer Integrated Manufacturing, 11(6), 538–547 (1998).
45 Sreevalsan, P.C., & Shah, J.J., Unification of Form Feature Definition Methods, Proceedings of the IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design, 83–106 (1992).
46. Staley, S.M., Henderson, M.R., & Anderson, D.C., Using Syntactic Pattern Recognition to Extract Feature Information from a Solid Geometric Database, Computers in Mechanical Engineering, 2(2), 61–66 (1983).
47. Tseng, Y.J., & Joshi, S.B., Recognizing of Interacting Rotational and Prismatic Machining Features from 3D Mill-Turn Parts, International Journal of Production Research, 36(11), 3147–3165 (1998).
48. Venkataraman, S., Sohoni, M., & Kulkarni, V., A Graph-Based Framework for Feature Recognition, Appearing in ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, 194–205 (2001).
49. Zeid, I., Mastering CAD/CAM. McGraw-Hill Higher Education (2004).
50. Zhang, G.X., Liu, S.G., Ma, X.H., & Wang, Y.Q., Toward the Intelligent CMM, CIRP Annuals, Manufacturing Technology, 51(1), 437–442 (2002).
51. Zhao, Z., Ghosh, S.K., & Link, D., Recognition of Machined Surfaces for Manufacturing Based on Wireframe Models, Journal of Materials Processing Technology, 24(1), 137–145 (1990).

Chapter 7
Rapid Manufacturing

Bahram Asiabanpour[1], Alireza Mokhtar[2], and Mahmoud Houshmand[2]

Abstract This chapter defines rapid manufacturing (RM) as a technique for manufacturing solid objects by the sequential delivery of energy and/or material to specified points in space. Current practice is to control the manufacturing process by using a computer-generated mathematical model. This chapter compares the large speed and cost advantages of RM to alternative polymer or metal manufacturing techniques such as powder metallurgy manufacturing or die casting. Moreover, the RM as an application of solid freeform fabrication for direct manufacturing of goods is addressed. Unlike methods such as computer numerical control (CNC) milling, these techniques allow the fabricated parts to be of high geometric complexity.

7.1 Rapid Manufacturing

Rapid manufacturing (RM), also known as direct manufacturing/direct fabrication/digital manufacturing, has been defined in various ways. One widely accepted definition is "the use of an additive manufacturing process to construct parts that are used directly as finished products or components" [1]. If this process occurs in the R&D stage, it is called *rapid prototyping* (RP) [2]. In the USA, the term "Solid Freeform Fabrication" is preferred to rapid prototyping or RM [3]. According to Plesseria et al., starting with a 3-D CAD part model, this technique converts the model to a series of layers using software; these layers are transferred to the building machine in which they are "printed" from the material by different processes. After printing one layer, a new layer of material is deposited and so on. Postprocessing treatment may be supplemented [4]. In other words, RM is the fabrication of parts or components using additive manufacturing technologies. The part is shaped layer-by-layer and could be used as a functional product. In this technique, the removal

[1] Ingram School of Engineering, Texas State University, San Marcos, Texas, USA

[2] Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

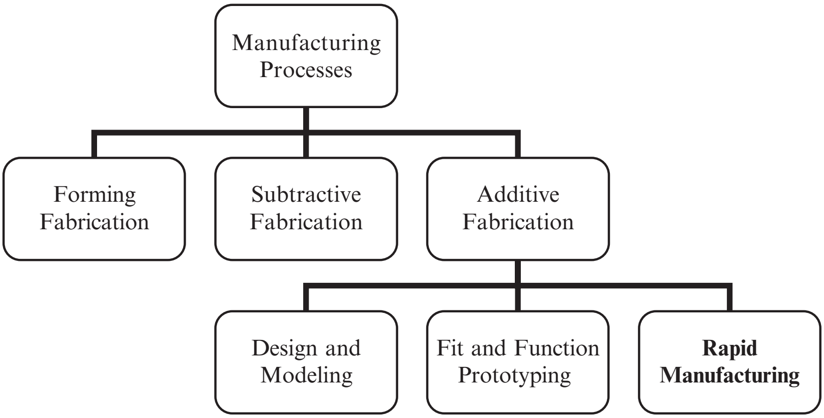


Fig. 7.1 Manufacturing processes tree and rapid manufacturing (RM) position [6]

cutting tools are not required to produce the physical products and the parts with complicated geometry may be fabricated [5]. RM is sometimes considered one of the RP applications. However, there is no clear distinction among definitions. RM's outputs are often usable products. Stereolithography (SLA), laser sintering (LS), and fused deposition modeling (FDM) are some of the typical RM machines. As shown in Fig. 7.1, RM can also be expressed as a branch of additive fabrication, which refers to the technologies employed to create physical models, prototypes, tools, or finished parts using 3-D scanning systems.

7.1.1 Applications of Rapid Manufacturing

RM is widely used for both large- and small-scale products and components for a variety of applications in many different fields. The main applications of the RM can be categorized as follows [7].

7.1.1.1 Tooling and Industrial Applications

Fabrication of metal casting and injection mold has been one of the main applications of RM in recent years, and has been addressed in the literature [8–11].

7.1.1.2 Aerospace

RM products have found efficient applications in spacecraft structures (mirrors, structural panels, optical benches, etc.). They are made up of different titanium and aluminum alloys (with granulated powder) and other materials such as silicon carbide, metal ceramic composites (SiC/Alu, ferrous materials with SiC), and carbon fiber reinforced polymer (CFRP) [12].

7.1.1.3 Architecture and Construction

In today's construction, CAD/CAM technology, industrial robots, and machines which use direct numerical control and RM open up the architectural possibilities. The Buswell's research into the use of mega-scale RM for construction [13], Pegna's investigation of solid freeform construction [14], and Khoshnevis's development of a new layer-by-layer fabrication technique called contour crafting [15] are some of the attempts to apply RM in architecture and construction industry.

7.1.1.4 Military

The production costs of military complex airframe structures were notably reduced where direct metal deposition, an RM technique, was applied to build limited number of metallic parts [16].

7.1.1.5 Medical Applications

Medical application of RM is mainly due to its capability to build uniquely shaped products having complex geometry. Medical products like implants, dentures, and skull and facial bones are the components that often vary from one person to another. Some examples are (1) custom-made orthodontic appliances production using proprietary thermoplastic ink jetting technology and (2) ear prostheses and burn masks for patients by using thermo-jet printing and FDM [17, 18]. Other applications have been reported to manufacture patient-specific models (lead masks) as well as protective shields in cancer treatment by using 3-D photography and metal spraying technology [19]. Forming a prosthetic glove of nontoxic materials to cover a patient's damaged hand is another medical application [20, 21].

7.1.1.6 Electronics and Photonics

A new RM methodology based on a direct-write technique using a scanning laser system to pattern a single layered SU-8 for fabrication of embedded microchannels has been reported by Yu et al. [22]. Nijmeijer et al. explain how microlaminates, which are widely used in ceramic capacitors of electronic devices, are being made by an RM process: centrifugal injection casting (CIC) [23].

7.1.2 Rapid Manufacturing's Advantages and Disadvantages

RM conducted in parallel batch production has a large advantage in speed, cost, and quality over alternative manufacturing techniques such as laser ablation or die casting. RM changes the cost models in conventional supply chains and has a key role in producing and supplying cost-effective customized products [24]. Consequently, RM's popularity is growing on a daily basis. According to a Wohlers Associates survey, RM's applications in additive processes grew from 3.9% in 2003 to 6.6% in 2004 and 8.2% in 2005 [25].

7.1.2.1 Advantages

RM's advantages can be studied from several design-related perspectives.

- **Design complexity:** One major benefit of the additive manufacturing processes is that it is possible to make parts of virtually any geometrical complexity at no extra cost while in every conventional manufacturing technique there is a direct link between the complexity of a design and its production cost. Therefore, for a given volume of component, it is possible to get the geometry (or complexity) for "free," as the costs incurred for any given additive manufacturing technique are usually determined by the time needed to build a certain volume of part, which, in turn, is determined by the orientation that the component is built in [26].
- **Design freedom:** The advent of RM will have profound implications for the way in which designers work. Generally, designers have been taught to design objects that can be made easily with current technologies—this being mainly due to the geometry limitations of the available manufacturing processes. For molded parts, draft angles, constant wall thickness, location of split line, etc. have to be factored into the design. Because of the advancements in RM, geometry will no longer be a limiting factor in design [26].
- **New design paradigm:** RM has simplified the interaction between mechanical and aesthetic issues. With current RM capabilities, industrial designers can design and fabricate the parts without the need to consider issues such as draft angle and constant wall thickness that are needed for processes such as injection molding. Similarly, mechanical designers are able to manufacture any complexity of product they require with minimum education in the aesthetic design field [26].

7.1.2.2 Disadvantages

Like any other immature technology, RM has drawbacks and limitations which preclude its widespread commercial application. Some main disadvantages are as follows:

- **Material cost:** Today, the cost of most materials for additive systems is about 100–200 times greater than that of those used for injection molding [27].
- **Material properties:** Thermoplastics from LS have performed the best for RM applications. However, a limited choice of materials is available. Actually, materials for additive processes have not been fully characterized. Also, the properties (e.g., tensile property, tensile strength, yield strength, and fatigue) of the parts produced by the RP processes are not currently competitive with those of the parts produced by conventional manufacturing processes. Ogando reports that the properties of a part produced by the FDM process with acrylonitrile butadiene styrene (ABS) material are about 70–80% of a molded part. In contrast, in some cases, results are more promising for the RP processes. For example, the properties of the metallic parts produced by the direct metal laser sintering (DMLS) process are "very similar to wrought properties and better than casting in many cases." Also, in terms of surface quality, even the best RM processes need secondary machining and polishing to reach acceptable tolerance and surface finish [28].
- **Support material removal:** When production volumes are small, the removal of support material is usually not a big issue. When the volumes are much higher, it becomes an important consideration. Support material that is physically attached is of most concern.
- **Process cost:** At present, conventional manufacturing processes are much faster than additive processes such as RM. Based on a comparison study by the Loughborough University, the SLA of a plastic part for a lawn mower will become economical at the production rate of 5,500 parts. For FDM, the break-even point is about 6,500 parts. Nevertheless, injection molding is found to be more economical when larger quantities must be produced [29].

7.2 Rapid Manufacturing Errors

One of the main challenges in RP and RM is deviation from the CAD part model. The accuracy will be obviously enhanced by increasing the number of layers (decreasing the layer thickness) but the manufactured part is never identical to its CAD file because of the essence of layer-by-layer fabrication. Three major RM errors are as follows.

7.2.1 Preprocess Error

While converting a CAD to standard tessellation language (STL) format (as machine input), the outer surface of the part is estimated to some triangles and this estimation causes this type of error, especially around the points with higher curvature (lower radius). Meshing with smaller triangles may diminish this error. However, it requires more time to process the file and a more complicated trajectory for laser (Fig. 7.2).

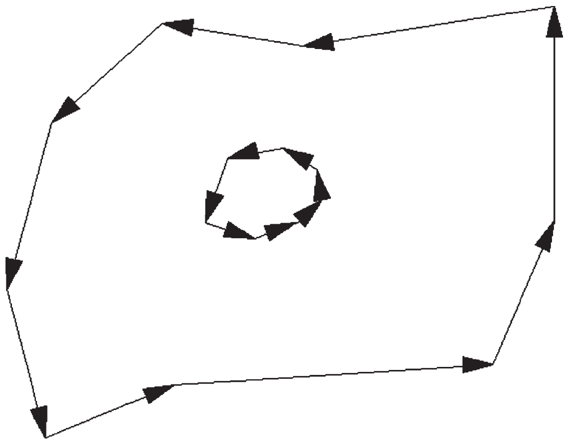


Fig. 7.2 More triangles result in more edges in each layer [*30*]

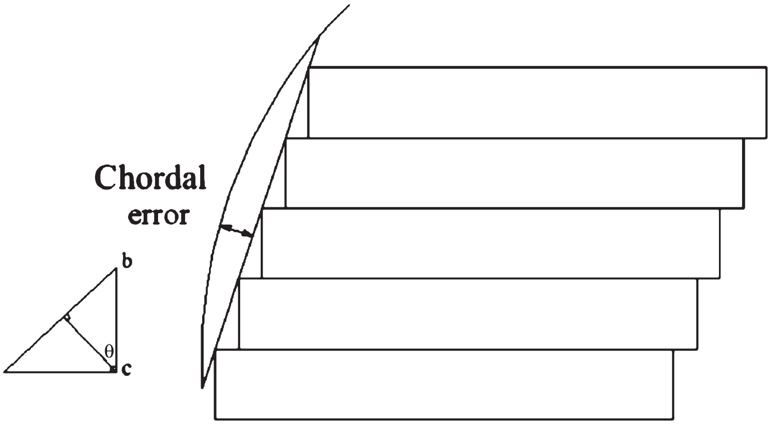


Fig. 7.3 This error can be estimated as *bc cos θ* [*30*]

7.2.2 Process Error

As shown in Fig. 7.3, slicing the part causes a new type of error (chordal error) while building the part layer-by-layer. This error depends on the layer thickness and the average slope angel of the part [30].

To minimize bc cos θ, the thickness of layer, bc, in the curved area should be decreased and it leads to lower step-stair effect and a more accurate product. Poor laser scanning mechanism may prompt another error during the process. The accuracy of laser beam emission and its angle with the part surface affect product quality so to make it smaller, the equipments and tools should be carefully inspected.

7.2.3 Postprocess Error

Two other types of errors that arise after the process are *shrinkage* and *warpage* errors. The product usually shrinks while cooling and this makes a deviation from its original design. Overestimation of CAD file at the design stage regarding the material properties and heating factors can reduce such an error. Warpage error is due to disparate distribution of heat in the part. Thermodynamic and binding force models are usually required to estimate and obviate this error [30].

7.3 Computer-Aided Rapid Manufacturing

Computer technology has served RM in many different aspects. In this section, tool path generation for RM by different CAD formats and computer-aided process selection are explained. The majority of the RP and RM processes use STL CAD format to extract the geometrical data of the model. The tool path generation from the STL file is explained in detail. Drawing exchange format (DXF) CAD file format is very complex and is mainly used either for data exchange between CAD software or tool path generation for the computer numerical control (CNC) machining process. However, the tool path generation method that is explained in this section can be an alternative approach for RM processes. Information about the standard for the exchange of product model data (STEP), an increasingly popular CAD format, is provided in this section.

7.3.1 Path Generation by Use of Drawing Exchange Format File

- **What is a DXF file**: The DXF is a CAD data file format developed by Autodesk to transfer data between AutoCAD and other programs. Each of the eight sections of a DXF file contains certain information about the drawing: HEADER section (for general information about the drawing), CLASSES section (for application-defined classes whose instances appear in the other sections), TABLES section (for definitions of named items), BLOCKS section (for describing the entities comprising each block in the drawing), ENTITIES section (for the drawing entities, including any block references), OBJECTS section (for the data that apply to nongraphical objects, used by AutoLISP and ObjectARX applications), and THUMBNAILIMAGE section (for the preview image for the DXF file) [31].
- **DXF for tool path generation:** Because of the complexity of the DXF files, generating a tool path from these files for any automated machining or fabricating process is very difficult. In such a tool path generation mechanism, geometrical data need to be identified and extracted from many other data available in a DXF file that are not useful for a tool path. Data can be extracted from models

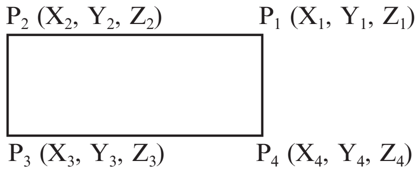


Fig. 7.4 DXF file data storage format for a 2.5-D object

that are designed as two-dimensional (2-D or wireframe) or two and half dimensional (2.5-D or surface) objects. Extracting data from DXF files containing 2.5-D objects is more complex than doing so from DXF files containing 2-D objects because for 2-D objects, the geometrical data are stored in the form of the object end points (e.g., start and end points of a line) while for 2.5-D objects, the geometrical data of a model (e.g., the surface of a sphere) are stored in the form of many small rectangles. In such case, for each rectangle *x*, *y*, and *z* coordination of all four corners are stored (Figt. 7.4).

- **DXF file generation:** Figure 7.5 illustrates the general steps to create a DXF file in the AutoCAD environment. As shown in this figure, after drawing the boundary of the object in wireframe format, a surface is generated to cover the entire outer boundary of the object. Then, after separating each face from the object, the entire data are stored in a DXF file. AutoCAD provides users with the flexibility to set a desirable number of meshes of the surface modeling in both horizontal and vertical directions.
- **Generating tool path from DXF files**: In automated controlling of the tool in any machine that uses the tool path, regardless of the process or the machine type, geometrical data need to be extracted from the CAD file. Then, the tool path is calculated and generated on the basis of the geometry of the model as well as process and tool specification. A tool in here can be a milling machine tool, a laser cutter head, a welding machine gun head, or an extruder nozzle. Figure 7.6 shows the general steps of generating tool path from a DXF file [32].

As shown in Fig. 7.6, in addition to geometrical data, the desired quality of the final part affects the tool path output. Figure 7.7 illustrates two different tool path configurations for the same CAD data.

- **Tool radius compensation:** If the tool radius is considered for the tool path generation, which is a must for almost all path-based processes, then the complexity of the tool path generation process becomes more complicated. In the tool radius compensation, the tool path is calculated for the center of the tool (not for the touching point of the tool and the part). Therefore, in the tool path generation process, both the curvature of the object at the tool and object touching point and the tool specification (size and shape) are affecting the center of the tool coordination. The curvature of the object at any point is shown by the normal vector. A normal vector is a vector (often unit) that is perpendicular to a surface (Fig. 7.8).

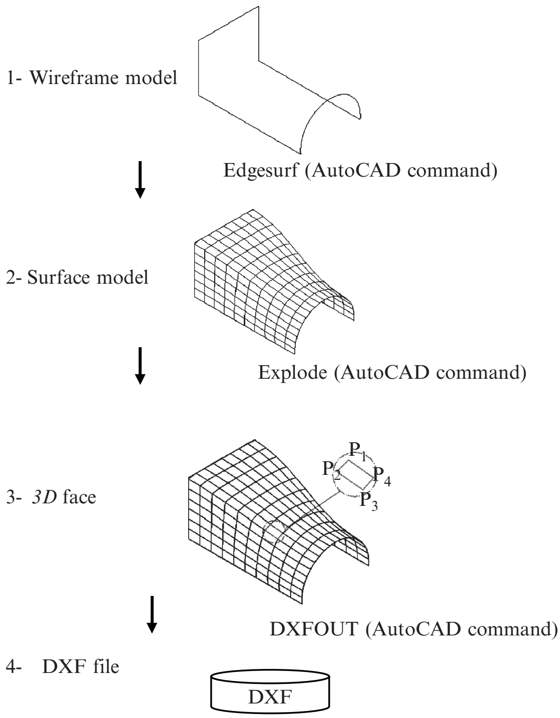


Fig. 7.5 DXF file generation procedure for a 2.5-D object in Auto-CAD environment

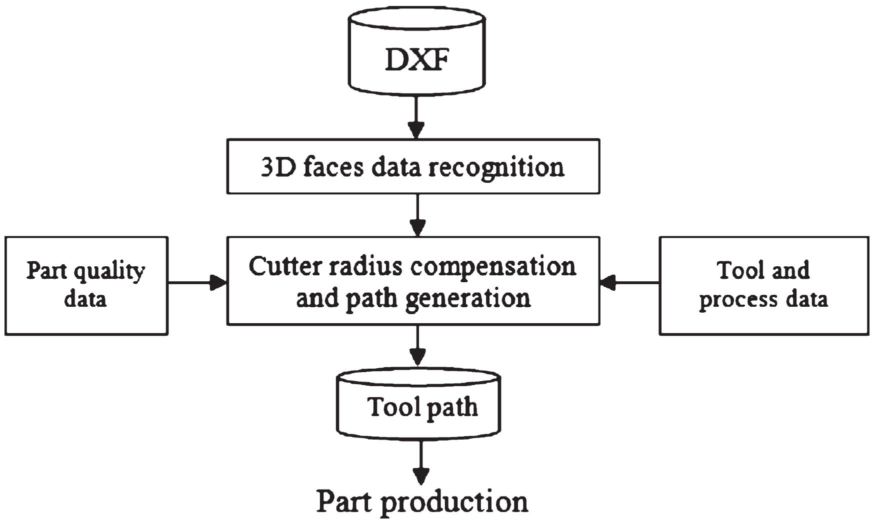


Fig. 7.6 Tool path generation from DXF file

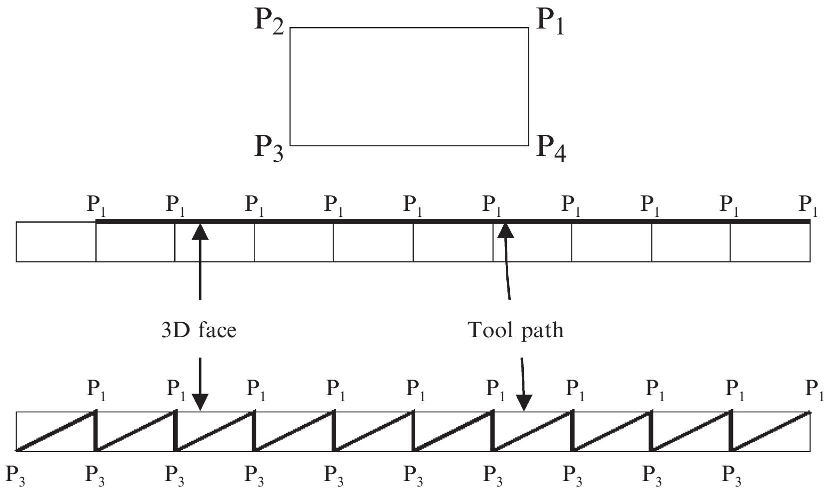


Fig. 7.7 Two different tool path configurations for the same CAD data

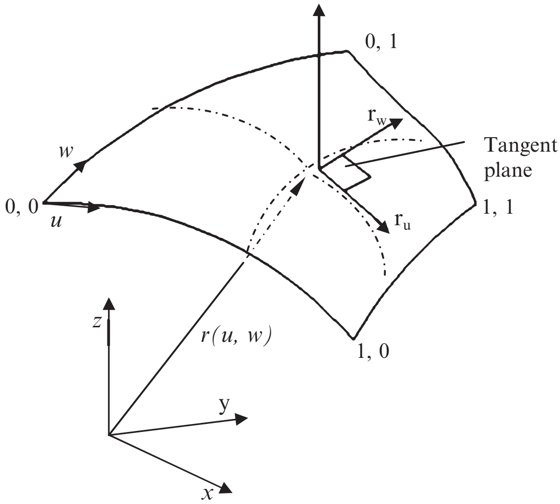


Fig. 7.8 A surface and its normal vector

Specifications of the tools may cover a variety of shapes and sizes. Three of the most common geometries of the tools are shown in Fig. 7.9. Most of the tools for machining and fabricating processes such as milling, extruding, welding, and laser beam are similar to one of these three geometries.

- **Flat end tool**: To calculate the position of the tool center for the tool path, the geometrical data of four corners of each rectangular 3-D face is used. In this process, the unit normal vector of the 3-D face is calculated, Eqs. 7.1–7.3, and

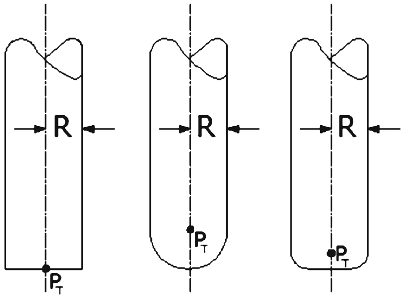


Fig. 7.9 Three common geometries for tools: 1 – flat end, 2 – ball end, and 3 – tip radius

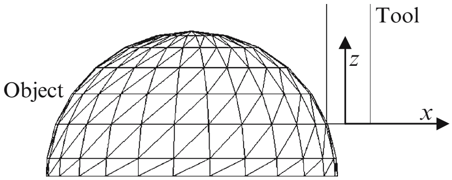


Fig. 7.10 Flat end tool

then it is used to determine the tool center position, Eqs. 7.4–7.6). In the flat end tools, the tool center is located at the same height (z level) as the tool and object touching point. This fact simplifies the calculation for the z level of the tool position. (Fig. 7.10).

$$\text{Normal vector} = r_u \times r_w = \begin{vmatrix} 1 & 1 & 1 \\ X_u & Y_u & Z_u \\ X_w & Y_w & Z_w \end{vmatrix} \tag{7.1}$$

$$= \left[\left(Y_u Z_w - Z_u Y_w \right) \quad \left(Z_u X_w - X_u Y_w \right) \right]$$

$$= \left[A \quad B \quad 0 \right]$$

$$r_u \,.\, r_w = (A^2 + B^2)^{0.5} \tag{7.2}$$

$$\text{Unit normal vector} = N_1(U,W) = \left[\frac{A}{\left(A^2+B^2\right)^{0.5}} \; \frac{B}{\left(A^2+B^2\right)^{0.5}} \; 0 \right] \tag{7.3}$$

- Tool center:

$$X_a = X_1 + R\frac{A}{\left(A^2+B^2\right)^{0.5}} \tag{7.4}$$

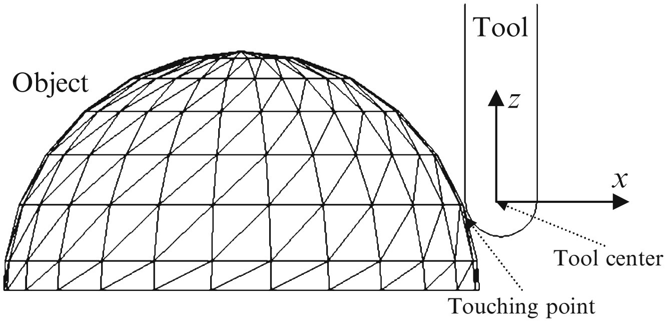


Fig. 7.11 Ball end tool

$$Y_a = Y_1 + R\frac{A}{\left(A^2 + B^2\right)^{0.5}} \tag{7.5}$$

$$Z_a = Z_1 \tag{7.6}$$

- **Ball end tool:** The calculation of tool center position for ball end tools is very similar to that for flat end tools. The only difference is the z level of the tool center that, similar to x and y coordination, needs to be determined by the unit normal vector of the 3-D faces (Fig. 7.11).

$$N_2(U,W) = \left[\frac{A}{\left(A^2+B^2+C^2\right)^{0.5}} \; \frac{B}{\left(A^2+B^2+C^2\right)^{0.5}} \; \frac{C}{\left(A^2+B^2+C^2\right)^{0.5}}\right] = N_2\left(n_x\, n_y n_z\right) \tag{7.7}$$

$$P_a = P_1 + R \cdot N_2\left(n_x \;\, n_y \;\, n_z\right) \tag{7.8}$$

- Tool center:

$$X_a = X_1 + R\frac{A}{\left(A^2 + B^2 + C^2\right)^{0.5}} \tag{7.9}$$

$$Y_a = Y_1 + R\frac{B}{\left(A^2 + B^2 + C^2\right)^{0.5}} \tag{7.10}$$

$$Z_a = Z_1 + R\frac{C}{\left(A^2 + B^2 + C^2\right)^{0.5}} \tag{7.11}$$

- **Tip radius tool**: The geometry of this type of tool is the combination of the last two tools (Fig. 7.12). The center part of this tool is a flat end tool while the edge

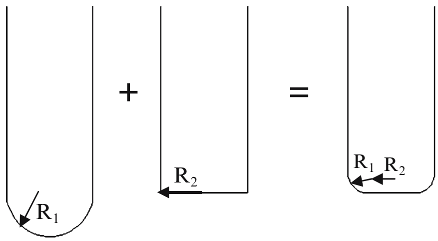


Fig. 7.12 Geometry of the tip radius tool

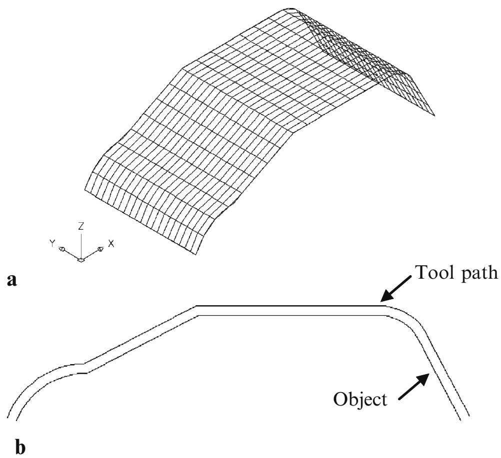


Fig. 7.13 **a** A meshed CAD object and **b** the position of the object and tool path

of the tool has a fillet. Therefore, the position of the tool center is determined on the basis of the above two tools' calculations.

$$P_a = P_1 + R_1 N_1 + R_2 N_2 \tag{7.12}$$

- Tool center:

$$X_a = X_1 + R_1 \frac{A}{\left(A^2 + B^2 + C^2\right)^{0.5}} + R_2 \frac{A}{\left(A^2 + B^2\right)^{0.5}} \tag{7.13}$$

$$Y_a = Y_1 + R_1 \frac{A}{\left(A^2 + B^2 + C^2\right)^{0.5}} R_2 \frac{B}{\left(A^2 + B^2\right)^{0.5}} \tag{7.14}$$

$$Z_a = Z_1 + R_1 \frac{C}{\left(A^2 + B^2 + C^2\right)^{0.5}} \tag{7.15}$$

Figure 7.13 illustrates a meshed CAD object (a) and the position of the object and tool path. At the end, it is necessary to mention that because of the complexity and limitations of the DXF files, it is very uncommon to use DXF to produce a tool path for RP, RM, or even 3-D and 2.5-D CNC machine code. Its application in CAM is usually limited to 2-D applications such as drilling.

7.3.2 *Path Generation by Use of STL File*

Every RM and RP system has its own specifications. The part boundary form, part filling method, and part separation from the surrounding material determine the tool path pattern for every layer. This tool path pattern could be a robotic movement in *XY* plane for FDM or contour crafting machines, a laser pattern for material solidification in SLA and selective laser sintering (SLS) machines, or a laser cutter pattern for a Laminated Object Manufacturing (LOM) machine. These processes require different tool path pattern generation strategies.

Therefore, unlike CNC standard tool path files (e.g. APT and G-Code), there is no standard tool path file for RP systems. Therefore, most of the new RM and RP require a new tool path generator or modification to the previous systems.

In this section, a tool path generation for the selective inhibition of sintering (SIS) process is presented. The software that is developed for this can be modified and adjusted for many other RP and RM processes. This system uses STL files with the ASCII format as input and works in two steps (Fig. 7.14).

7.3.2.1 Step 1—Slicing Algorithm

In Step 1, the STL file is read as input. Slice files are then generated by executing the slicing algorithm. Only the intersection of those facets that intersect current $Z = z$ is calculated and saved. In this step, one facet is read at a time from the STL file. Then the intersection lines of this facet with all *XY* planes for $Z_{min} \leq z \leq Min\{ Z_{max}, Max\{ z_A, z_B, z_C\}\}$ are calculated. The intersection lines are stored in the specified file for the associated z increment. This results in one intersection line on

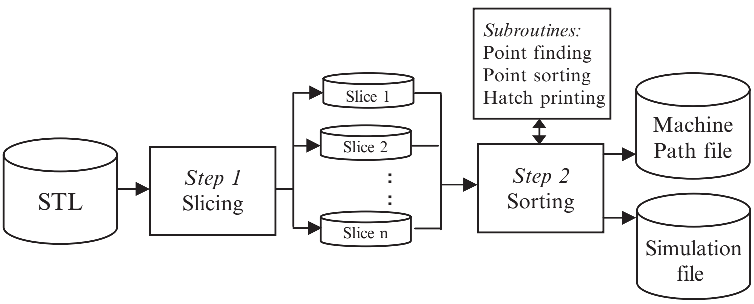


Fig. 7.14 The two steps of slicing and tool path generation

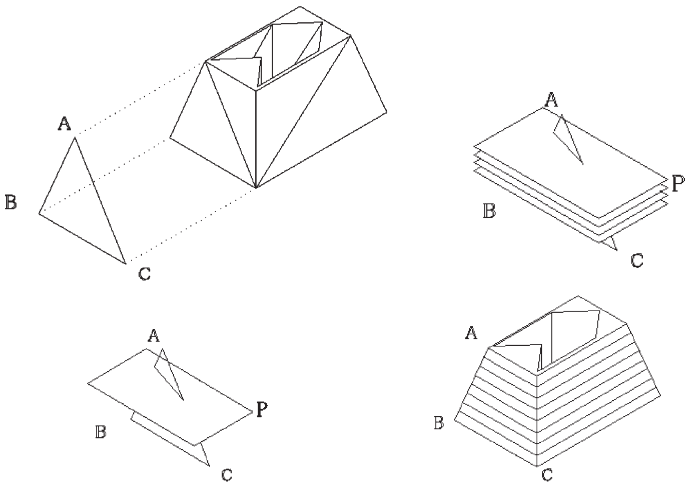


Fig. 7.15 Slicing algorithm steps

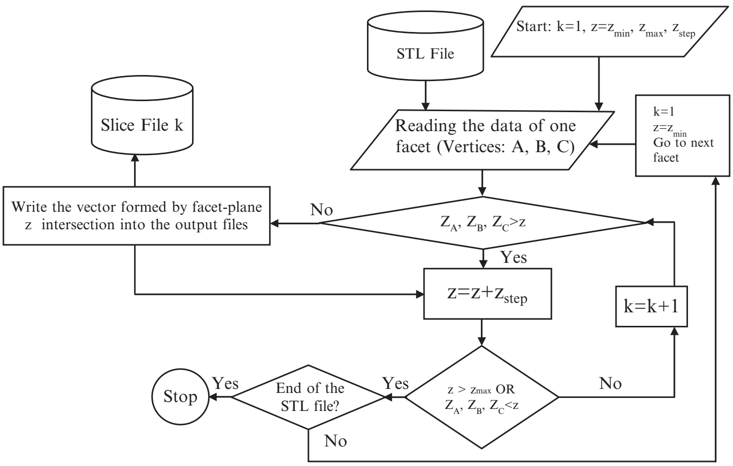


Fig. 7.16 The slicing algorithm

each *XY* plane. By repeating this process for all facets, a set of slices is generated. This algorithm saves the data of only one facet in the computer memory; therefore only a small amount of computer memory is needed, and there is no practical limitation on the model size. In this step, each slice is saved in a separate file on the disk. This guarantees that Step 2 is run much faster than when all slices are saved in a single file. The example shown in Fig. 7.15 illustrates the slicing algorithm and Fig. 7.16 shows the flowchart of the slicing algorithm.

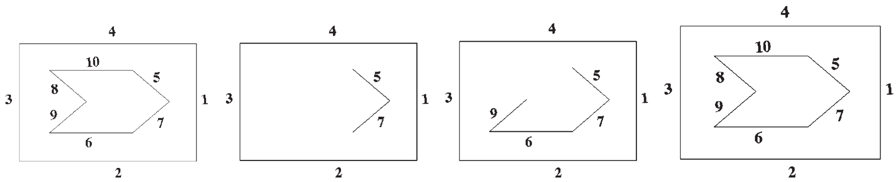


Fig. 7.17 Tool path generation steps

7.3.2.2 Step 2—Tool Path Generation

After the completion of the slicing process, a set of vectors becomes available in each z increment. These vectors are not connected and are not in sequence. In the tool path generation process, the software starts from one vector and tries to find the next connected vector to this vector. Then it does the same for the newly found vector until it reaches the start point of the first vector (in the closed loop cases) or finds a vector with no leading attachment (in faulty STL files containing disconnections). To sort the vectors, the algorithm reads one vector at a time from a slice file and writes it to another file. This file is either a path file, when one vector is connected to the previous vector, or a temp file, when the vector is not connected to the previous vector. Therefore, the sorting process does not need a large amount of memory to sort the data, and there is no limitation on the number of vectors in a slice and on input file size. In addition, unlike many other slicing algorithms that cannot handle disconnections caused by faulty facets [33], this algorithm can generate a tool path even with disconnection errors in the STL file. At disconnection instances the system sends a message to a log file and turns the printer off and starts from a new vector. In either case, the printer is turned off and the system starts printing from another start point. Also for each selected vector, the possibility of hatch intersection points is investigated.

At the end of the path generation process for one slice, the hatch intersection points are sorted and written into a tool path file. After the arrangement of all vectors in one slice (z increment), the process starts arranging the vectors of the next slice. This process continues until all vectors in all slices are sorted. The diagram in Fig. 7.17 and the flowchart in Fig. 7.18 represent the tool path generation algorithm.

7.3.2.3 Implementation

Slicing and tool path generation algorithms have been implemented in the C programming language. The software has been successfully tested for several medium and large STL files up to 200 MB on different PCs and laptops. Figures 7.19 and 7.20 show the algorithm implementation as presented by the path simulation module of the system [34, 35].

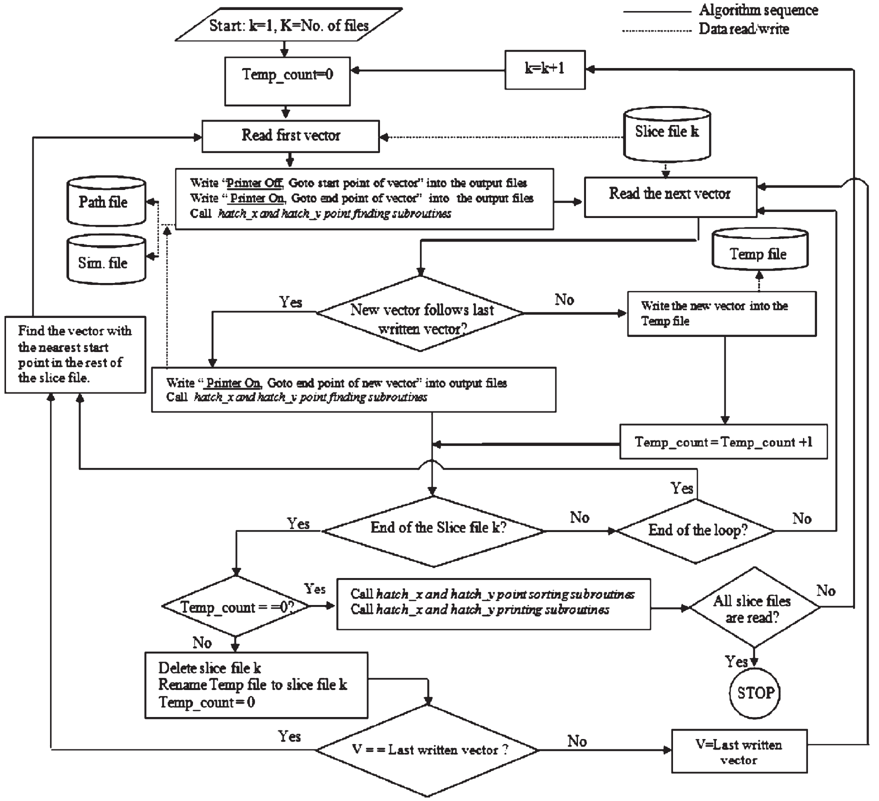


Fig. 7.18 The tool path generation algorithm

7.3.3 Path Generation by Use of STEP File

STEP, introduced by PDES, Inc., is a neural format for exchanging product data among all the people or organizations who contribute to marketing, design, manufacturing, and other activities in the product life cycle. STEP (identified as ISO 10303) makes an independent platform to access, share, and manipulate the product information of its life cycle.

There are other types of product data exchange (PDE) models like IGES, DXF, DWG, VHDL, and STL which are used by many industries all over the world but STEP is known as a superior standard and is becoming highly popular over the other formats such as IGES [36].

7.3.3.1 Application Protocols

STEP is developed by a specific language of EXPRESS and includes a number of manageable and functional sections referred to as application protocol (AP). Each

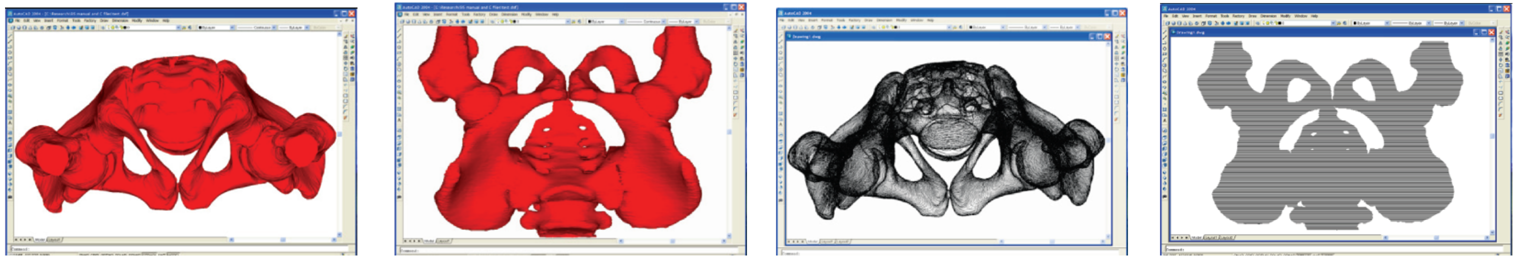


Fig. 7.19 Visualization of models (up) and slices (bottom)

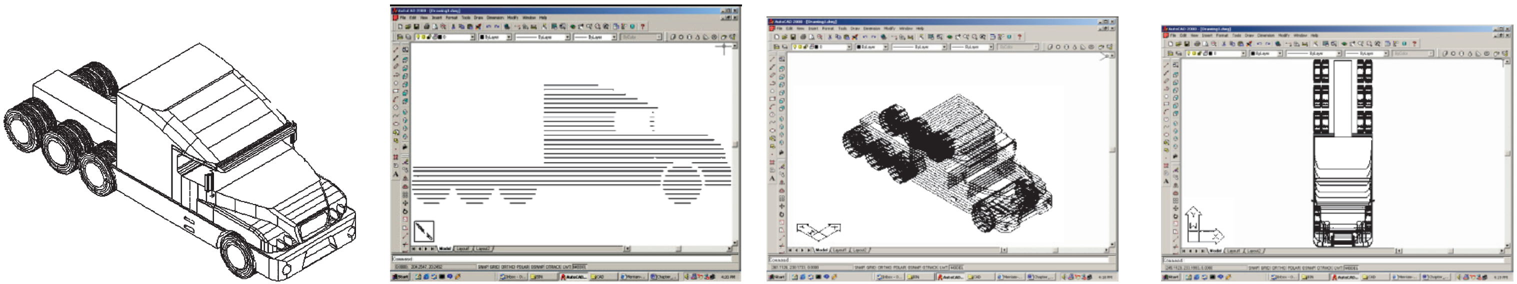


Fig. 7.20 Visualization of CAD file and the tool path in the Auto-CAD environment

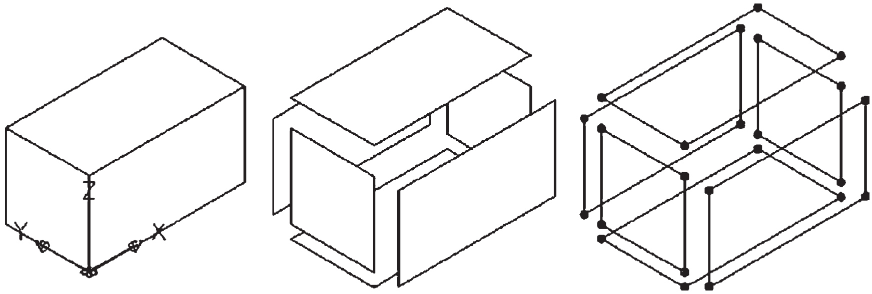


Fig. 7.21 B-rep model of a simple cube: faces, edges, and vertices depicted in the model

AP is in charge of an application of STEP. For example, AP224 or ISO 10303–224 are mechanical product definitions for process planning using machining features; AP203 is configuration-controlled 3-D designs of mechanical parts and assemblies. For the time being, only 22 APs have been approved as International Standards; however, it is expected that there will be hundreds of APs in the future.

STEP file contains all the required information for design, process planning, and other downstream activities; the implementation of STEP tools depends on the requirements and problem area in the organization. For process planning, a mechanical part based on its machining features, AP224, will be applicable as it classifies the features and can store and manage the data needed for each feature [37, 38].

7.3.3.2 Boundary Representation Model

Some APs contain boundary representation (B-rep) models for mechanical parts. For every 3-D part, B-rep expresses the surface boundary, which consists of the part's geometrical and topological information. Vertices, edges, and faces information is also represented by a B-rep model [39] (Fig. 7.21).

7.3.3.3 Using STEP for Tool Path Generation

No direct usage of STEP file to generate the tool path in RM has been reported. But as indicated, some of the application protocol contains B-rep information for the products. This information model can be extracted and used for tool path generation. Consider a part meshed with triangles and sliced in STL file format is replaced with a part having information of all the edges, faces, and vertices. Laser trajectory will be obtained by introducing the points of outer surfaces. This may not be applicable only for free-form parts with complex geometry as their B-rep information is not complete on the surfaces (Fig. 7.22).

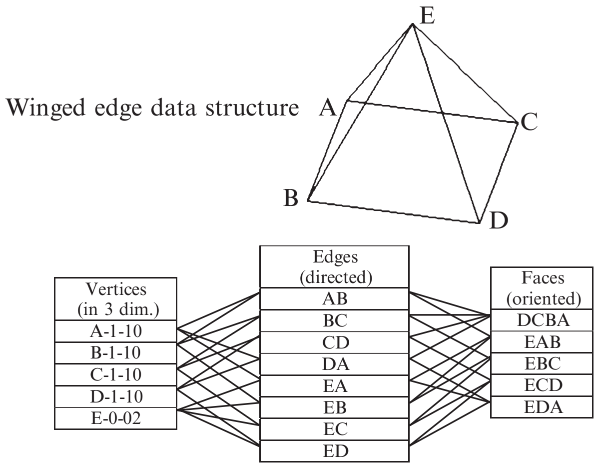


Fig. 7.22 Boundary representation (B-rep) of a simple part [40]

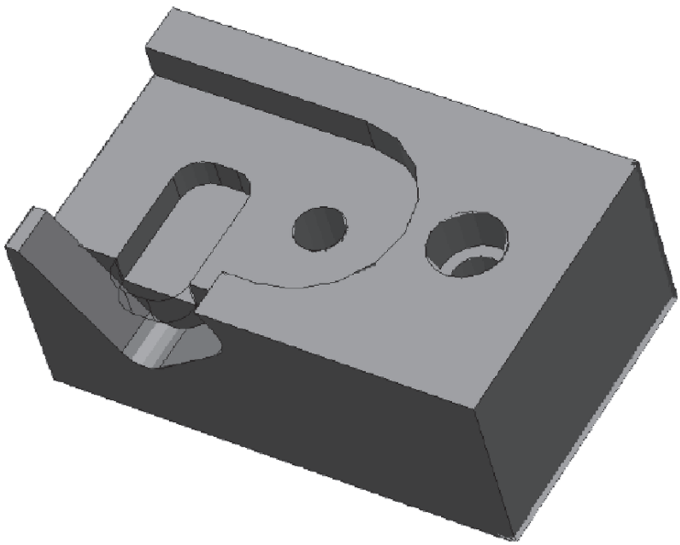


Fig. 7.23 A typical part to be rapid-manufactured

Even with a B-rep model, slicing is required to determine the exact intersection points and feed the control system of laser scanning. A sample CAD model and its STEP file are shown in Figs. 7.23 and 7.24, respectively.

AP240, numerical control process plans for machined parts (ISO 10303–240:2005), is a part of ISO 10303 that specifies information requirements for the exchange, archival, and sharing of computer-interpretable numerical control (NC) process plan information and associated product definition data but it does not

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('This is a test file'),'1');
FILE_NAME('test','2007-8-18',('Authors'),('Authors'),'NA','OK');
FILE_SCHEMA('ARM224');
ENDSEC;
DATA;
#1 = CARTESIAN_POINT((0.176777,0.176777,0.5));
#2 = VERTEX_POINT(#1);
#3 = CARTESIAN_POINT((0.,0.,0.5));
#4 = DIRECTION((0.,0.,–1.));
#5 = DIRECTION((0.176777,0.176777,0.));
#6 = AXIS2_PLACEMENT_3D(#3,#4,#5);
#7 = CIRCLE(#6,0.25);
#8 = EDGE_CURVE(#2,#2,#7,.T.);
#9 = ORIENTED_EDGE(*,*,#8,.F.);
#10 = EDGE_LOOP((#9));
#11 = FACE_BOUND(#10,.T.);
#12 = CARTESIAN_POINT((0.176777,0.176777,–0.5));
#13 = VERTEX_POINT(#12);
#14 = CARTESIAN_POINT((0.,0.,–0.5));
#15 = DIRECTION((0.,0.,1.));
```

Fig. 7.24 A piece of a STEP file

include specific machine tool controller codes. AP204, AP207, AP210, AP214, AP224, AP227, and AP240 are the applications which have boundary information of the corresponding parts and may be used for tool path generation.[41]

7.3.4 Rapid Manufacturing Process Selection and Simulation

Because of the inherent strengths and weaknesses of all RM processes, choosing the method best fitted to economical and technical objectives is a serious problem. This process selection is a multi-criteria decision making. Along with satisfying customer requirements, the functional product is expected to be manufactured to an adequate quality and at most reasonable cost in the shortest possible time. Other criteria such as product recyclability or serviceability may be considered.

One of the approaches for making such simulation in RP and RM uses computer software in order to create a virtual prototype. It is due to high material cost and prototyping of the real prototypes. Simulators help engineers to evaluate the performance of the process and minimize the number of repetitions to reach an appropriate prototype [42]. One successful attempt to introduce such simulators was made by Choi and Samavedam [43]. They developed a simulation software program to show the RM process layer-by-layer with SLS process. It shows also the relation between process parameters and time and accuracy of the product. The main advantages of running simulation in RP and RM are listed below:

- The cost of materials especially for some processes such as SLS is notable.
- The process of making the product is usually time-consuming.
- The energy consumption and equipment depreciation are high.
- The quantitative parameters of the prototype will be easily extracted.
- The product information may be shared with other persons or research centers [44].

VIRAPS (Virtual Rapid Prototyping System) is a simulation software program developed by Visual Basic and simulates some of the most common processes of RP and RM, like FDM, SLS, SLA, and LOM. Here we describe how VIRAPS works and the outcomes.

There are many criteria to consider in choosing the right RM process. Since it is a multi-criteria decision, an analytic hierarchy process (AHP) methodology could be applied to rank the most appropriate process according to the customer's criteria. Using software, RM processes are compared to each other on the basis of some common characteristics such as average time, cost, and quality of the finished products. The priority of these criteria is also determined by customer and the highest ranked RM process and machine is selected accordingly [45].

As a case study, SLS is considered for simulation. Suppose that a conic-shaped product is manufactured by SLS. The important inputs for this process are as follows:

- Part information (maximum dimensions, average slope, and layer thickness)
- Laser specification (laser type, laser power, beam diameter, and scan speed)
- Powder selection (here is steel-bronze)
- Setup time for each layer (the time required between creating each layer for setup)
- Cost parameters (direct cost, operation cost, and material cost)

The software also indicates the maximum and minimum allowed for some parameters such as scan speed or beam diameter and product weight according to the machine capabilities. Moreover, some parameters, such as type of laser, are automatically selected by the software [*46*] (Figs. 7.25 and 7.26).

The outputs are shown in Fig. 7.27. This product will be made in 6.5 h and with the accuracy ratio of 73 and efficiency of 47%. The total estimated cost is \$344 and the number of layers according to the best orientation of the part is 500. These results are calculated on the basis of input information. For instance, the time in SLS is formulated below [30]:

$$\text{Velocity}(v) = \frac{P_1(1-R)}{\rho d_b l_m \left[C_p(T_m - T_b)\right]} \tag{7.16}$$

where velocity (v) = velocity of laser (mm/s), P_1 = power of laser emission (watt), R = reflection ratio of laser reflector mirror, ρ = material density (g/mm^3), d_b = beam diameter (mm), l_m = layer thickness (mm), C_p = specific heat capacity (J/K), T_m = material melting point (K), and T_b = laser scanning time (s).

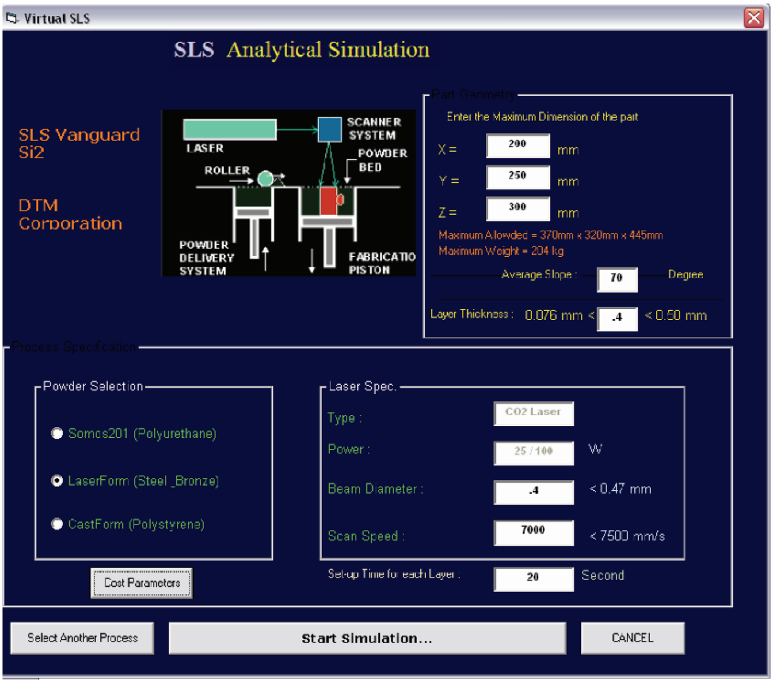


Fig. 7.25 The SLS analytical simulation interface

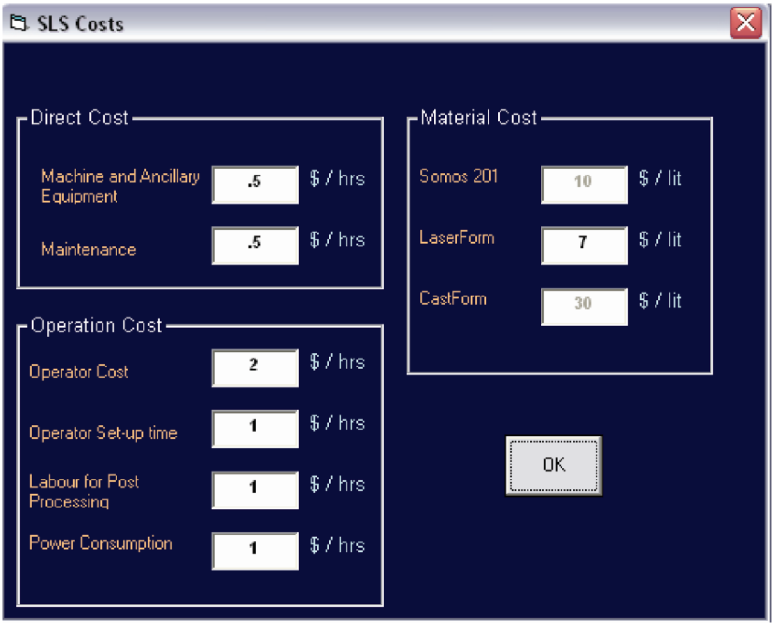


Fig. 7.26 Software interface for process cost items

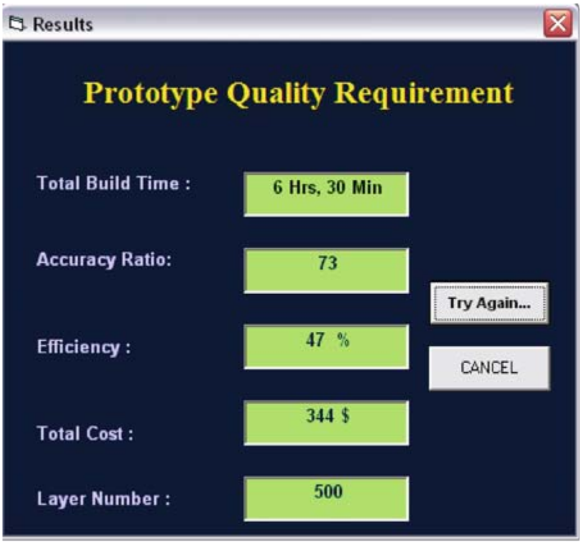


Fig. 7.27 Process simulation outcomes

On the one hand, the time required for building the whole part is the time taken for making all the layers and the time of setup for each layer. On the other hand, the time of scanning each layer, T_l, is $\frac{L_d}{L_v}$ in which L_d is the laser scanning distance and L_v is the laser scanning speed.

$$\text{Buildtime(total)} = \sum_{i=1}^{N} T_{li} + T_s N_l \tag{7.17}$$

where build time (total) = total time for building layers, T_{li} = scanning time for layer i, T_s = setup time for each layer, and N_l = number of layers.

Setup time is separately obtained by the following relations:

$$\text{Setuptime(Ts)} = T_{wd} + T_d + T_{wr} + T_h \tag{7.18}$$

where T_{wd} = time required for moving the part bed downward (s), T_d = time required for pouring a layer of material (s), T_{wr} = time required for moving the part bed upward (s), and T_h = time required for preheating the material (s).

Therefore, assuming each layer has thickness of l and the distance scanning of laser for layer i is d_{si}, for a part with height of h, we have

$$\text{Buildtime(total)} = \left(\frac{h}{l_m}\right) T_s + \frac{\sum_{i=1}^{N_l} d_{si}\left(\frac{l}{l_m}\right)}{L_V} \tag{7.19}$$

7.4 Rapid Manufacturing Prospects

Today, RM is widely used in some companies. However, because of the material and process limitations of the current RM processes because of the lack of familiarity with these machines, RM is not as popular as it should be. It is estimated that in the next 10–20 years, engineers will recognize the benefits of RM processes [28].

References

1. Dickens P., Research goals in rapid manufacturing. Metalworking Production 148:21–22, 2004..
2. Hopkinson N., Hague R., Dickens P., Rapid Manufacturing, (Abstract). Germany: Wiley-VCH., 2005.
3. Soar R.C., Gibb A.G.F., Thorpe A., Buswell R.A., Freeform construction: mega-scale rapid manufacturing for construction. Automation in Construction 16:224–231, 2007..
4. Rochus P., Plesseria J.Y., Van Elsen M., Kruth J.P., Carrus R., Dormal T., New applications of rapid prototyping and rapid manufacturing (RP/RM) technologies for space instrumentation. 61:352–359, 2007.
5. Hague R., Mansour S., Saleh N., Material and design considerations for rapid manufacturing. International Journal of Production Research 42:4691–4708, 2004.
6. Wohlers Associates Inc., What is Additive Fabrication? 2006.
7. Science group in groupsrv.com, http://www.groupsrv.com/science/index.php.
8. Nan Z., Gang W., Application of rapid prototyping techniques to rapid manufacturing of mould. Hebei Journal of Industrial Science & Technology 21:38–40, 2004.
9. Shi Y.-S., Cheng W., Huang N.-Y., Huang S.-H., Rapid manufacturing technology for casting mould. Zhuzao (Foundry) 54:382–385, 2005.
10. Li Y.-M., Hao Y., Huang N.-Y., Fan Z.-T., Dong X.-P., Liu H.-J. Casting-the key enabling technology promoting rapid manufacturing of metallic parts or moulds, Research Center of Die and Mould Technology, State Key Laboratory of New Non-ferrous Metal Materials, Lanzhou University of Technology.
11. Zhao J.F., Yu C.Y., Wu X.L., Jishu J.K.U., Zhang J.H., A new technology for rapid manufacturing of 3D moulds. Mechanical Science and Technology 20:419–420, 2001.
12. Plesseria J.Y., Van Elsen M., Kruth J.P., Carrus R., Dormal T., Rochus P., New applications of rapid prototyping and rapid manufacturing (RP/RM) technologies for space instrumentation 61:352–359, 2007.
13. Buswell R.A., Gibb A.G.F., Mega-scale Rapid Manufacturing for construction. Automation in Construction 16:24–31, 2007.
14. Pegna J., Exploratory investigation of solid freeform construction. Automation in Construction 5:427–437, 1997.
15. Khoshnevis B., Automated construction by contour crafting—related robotics and information sciences, Automation in Construction (special issue): The Best of ISARC 13:5–19, 2004..
16. Calder N.J., Place for rapid manufacturing in military airframe production. Materials Technology 15:34–37, 2000.
17. Product innovations, Rapid manufacturing produces custom lingual orthodontic appliances. Modern Casting 96:76, 2006.
18. Watson J., Rowson J.E., Holland J., Harris R.A., Williams D.J., Application of Rapid Manufacturing Techniques in Support of Maxillofacial Treatment: Evidence of the Requirements of Clinical Applications. Journal of Engineering Manufacture 219:469–475, 2005.
19. De Beer D.J., Truscott M., Booysen G.J., Barnard L.J., Van Der Walt J.G., Rapid manufacturing of patient-specific shielding masks, using RP in parallel with metal spraying. Rapid Prototyping Journal 11:298–303, 2005.

20. Alves N.M.F., Bartolo P.J.S., Ferreira J.C., Rapid manufacturing of medical prostheses. International Journal of Manufacturing Technology and Management 6:567–583, 2004.
21. Sanghera B., Naique S., Papaharilaou Y., Amis A., Preliminary study of rapid prototype medical models. Rapid Prototyping Journal 7:275–284, 2001.
22. Yu H., Balogun O., Li B., Murray T.W., Zhang X., Rapid Manufacturing of Embedded Microchannels from a Single Layered SU-8, and Determining the Dependence of SU-8 Young's Modulus on Exposure Dose with a Laser Acoustic Technique In: Proceedings of the 18th IEEE International Conference on Micro Electro Mechanical Systems, MEMS 2005, Miami—Technical Digest 654–657, 2005.
23. Biesheuvel P.M., Nijmeijer A., Kerkwijk B., Verweij H., Rapid manufacturing of microlaminates by centrifugal injection casting. Advanced Engineering Materials 2:507–510, 2000..
24. Tuck C., Hague R., The pivotal role of rapid manufacturing in the Production of Cost-Effective Customised Products. International Journal of Mass Customisation 1:360–373, 2006.
25. Wohlers T., "Viewpoint," *Time-Compression Technologies*, 2006.
26. Hague R., Design Opportunities with Rapid Manufacturing, Rapid Manufacturing Research Group, Wolfson School of Mechanical and Manufacturing Engineering., Loughborough University, Loughborough, UK.
27. Hague R., Mansour S., Saleh N., Material and design considerations for rapid manufacturing. International Journal of Production Research 42:4691–4708, 2004.
28. Ogando J., Rapid manufacturing's role in the factory of the future. Design News, 2007.
29. Tuck C., Hague R., Make or buy analysis for rapid manufacturing. Rapid Prototyping Journal 13:23–29, 2007.
30. Choi S.H., Samavedam S., Modeling and optimization of rapid prototyping, Computers in Industry 47:39–53, 2002.
31. Revisions to the DXF Reference, http://www.autodesk.com/
32. Asiabanpour B., CAMAIRIC: A Computer Software in Computer Aided Manufacturing In: The 6th Annual Industrial Engineering Conference. Tehran, Iran., 1999.
33. Leong K.F., Chan C.K., Ng Y.M., A study of stereolithography file errors and repair. The International Journal of Advanced Manufacturing Technology 407–414, 2004.
34. Asiabanpour B., Khoshnevis B., Machine path generation for the SIS process. Journal of Robotics and Computer Integrated Manufacturing 20:167–264.
35. Asiabanpour B., Khoshnevis B., Palmer K., Advancements in the selective inhibition of sintering process development. Virtual and Physical Prototyping Journal 1:43–52, 2006.
36. Mangesh P., Bhandarkar N.R., STEP-based feature extraction from STEP geometry for Agile Manufacturing, Computers in Industry, 41(1): 3–24, 2000.
37. SCRA, Step Application Handbook, ISO 10303 Version 3, 2006.
38. ISO 13030–224, Industrial Automation Systems and Integration; Product Data Representation and Exchange, Part 224. Application protocol: Mechanical Product definition for process plans using machining features, 2001.
39. Shapiro V., Donald L. Vossler, "What is a parametric family of solids?", Proceedings of the third ACM symposium on solid modeling and application Salt Lake City, Utah, ACM, New York, USA, 43–54, 1995, ISBN: 0-897-91-672-7.
40. Sequin C.H., Foundations of the computer graphics, http://www.cs.berkeley.edu/~sequin/CS184/IMGS/Breps.GIF
41. Kramer T.R., Huang H., Messina E., Proctor F.M., Scott H., A feature-based inspection and machining system. Computer-Aided Design 33:653–669, 2001.
42. Nee A.Y.C., Fuh J.Y., Miyazawa T., On the Improvement of the stereolithography (SL) process. Journal of Materials Processing Technology 113:262–268, 2001.
43. Choi S.H., Samavedam S., Modeling and optimization of rapid prototyping. Computers in Industry 47:39–53, 2002.
44. Choi S.H., Samavedam S., Visualization of rapid prototyping. Rapid Prototyping Journal 7:99–114, 2001.
45. Mokhtar A.R., Applying AHP Methodology to Select the Best Rapid Prototyping Technique In: The 32nd International Conference of Industrial Engineering, Ireland, Aug 2003.
46. Houshmand M., Mokhtar A.R., Analytical simulation of rapid prototyping process in a virtual environment. Journal of Science and Technology 28:86–91, 2005.

Chapter 8
Simulation-Based Optimization: A Case Study for Airline's Cargo Service Call Center

Tiravat Assavapokee[1] **and Ibrahim Mourtada**[2]

Abstract In this chapter, we introduce the basic concept of the simulation-based optimization and illustrate its usefulness and applicability for generating the manpower planning of airline's cargo service call center. Because of the continuous increase in oil prices, and combined with many other factors, the airline industry is currently facing new challenges to keep its customers satisfied. One of the most important drivers of the customer satisfaction is the customer service. The excellent customer service can give an airline company the edge over its competitors. Airline companies need to insure the appropriate level of staffing at their service call centers in order to maintain a high level of customer satisfaction with the appropriate level of the overall cost. With the high level of uncertainty in the customer demand and a number of complicated factors in the problem, it becomes necessary to apply the simulation-based optimization technique to help managers generate the efficient staffing policy for the airline's cargo service call center. In this work, the technique called reinforcement learning and Markov decision process are used to build and solve the mathematical model to determine the appropriate staffing policy at the airline's cargo service call center on the monthly basis. Simulation and optimization models are incorporated together so as to solve the overall problem. The results of the case study are thoroughly analyzed, discussed, and compared with the current staffing policies. All results illustrate the impressive performance of the recommended staffing policies over the current staffing policies.

8.1 Introduction

Simulation and optimization are clearly two of the most widely implemented operation research and management science techniques in practice. There have been several obstacles that limit acceptance and usefulness of simulation and optimization techniques in the past. For example, developing simulation and optimization

[1] Assistance Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77204, USA

[2] Manufacturing Engineer, Cameron Measurement Systems, Houston, Texas, 77032, USA

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,* doi: 10.2007/ 978-0-387-47321-5, © Springer Science + Business Media, LLC 2008

models for the large-scale real-world systems tends to be a very complex task. In addition, writing the computer code to execute and solve these models can also be another difficult and time-consuming process. Because of the recent advance in computer technology and recent development of modern simulation and optimization software, these obstacles have been significantly reduced (if not eliminated). Complex simulation and optimization models can now be developed much easier in recent years by utilizing the modern software packages that conveniently provide many of the features required to develop these models. In addition, one can now run the simulation and optimization models of complex systems much faster as computers become much more powerful.

Simulation refers to the broad concepts of operation research methodologies and techniques that imitate the behavior of the real-world system. Simulation is usually used to study and to improve the performance of the existing system or to design a new system under uncertainty without actually experimenting with the actual physical system. This feature makes simulation a very powerful operation research technique in practice because it is often too difficult and costly to perform physical studies on the actual system. Simulation is often used as evaluation tools to answer many important "what if" questions that decision makers may have about the system. For example, decision makers can use simulation to answer the question such as: "what would happen to the performance of the factory if the layout is changed?" Even though, simulation can be used to efficiently evaluate the system performance for a given solution, it is not capable of recommending the best solution for the complex decision-making problems by itself.

Optimization refers to the broad concepts of operation research methodologies and techniques that model the complex decision-making problems and recommend the best solution to these problems. Optimization is certainly one of the most powerful operation research techniques and it pervades the fields of engineering, science, and business. To apply optimization techniques, decision makers have to first formulate the mathematical models that capture the decision-making problems. The appropriate optimization techniques are then applied to find the solutions to these models. The general goal of optimization is to find the solution that yields the best value of the performance criterion under some restrictions in the decision-making problems. In many cases, the real-world decision-making problems cannot be fully represented by the mathematical models. Decision makers are often required to make a number of assumptions in order to construct the appropriate mathematical models for these problems. As a consequence of making these assumptions, the solution obtained by solving these mathematical models may not be fully applicable for some real-world decision-making problems.

Because of the usefulness and applicability of these two powerful operation research techniques, researchers and practitioners have always been trying to combine simulation and optimization techniques into an even more powerful decision-making tool. In fact, simulation-based optimization is not a new topic in operation research and management science literature. Since the time that the computer systems was invented and started making an impact on practical decision-making processes and scientific researches, researchers and practitioners have always wanted to optimize their decision-making systems by utilizing simulation models. However, it is only recently that remarkable success in realizing this objective has been seen in practice due to the dramatic increase in the power of computer systems over the years. Simulation-based optimization now has so much potential in almost every area of decision-making processes under uncertainty.

In Sect. 8.2, we briefly review the literature in the areas of simulation-based optimization and service call center staff planning. In Sect. 8.3, we discuss the basic reinforcement learning (RL) methodology. In Sect. 8.4, the case study from the real airline industry is discussed and the results from the case study are thoroughly analyzed and illustrated. We then conclude the chapter and give the summary of the overall work in Sect. 8.5.

8.2 Literature Review

In this section, we summarize a number of literatures related to simulation-based optimization techniques and service call center staff planning.

8.2.1 Literature Review for Simulation-Based Optimization and RL

As discussed earlier, simulation is a very powerful decision-making tool to perform "what if" analysis of the complex systems. Recent research discovery illustrates that simulation can be coupled with powerful optimization algorithms to solve complex real-world problems. The effectiveness of this approach depends on the quality of the simulation model that represents the real-world system. A high degree of understanding of the system being studied is often required. The book written by Gosavi [1] gives a good introduction to the topics of simulation-based optimization and RL techniques. Kleinman et al. [2] show that reductions in the cost of the airline delay can be obtained by using a simulation optimization procedure to process delay cost measurements. They discuss how the optimization procedure called simultaneous perturbation stochastic approximation (SPSA) can be used to process delay cost measurements from air traffic simulation packages and produce an optimal gate holding strategy. Rosenberger et al. [3] developed a stochastic model for airline operations by using a simulation package called SIMAIR. The developed model is not modular and does not allow other recovery procedures to be integrated. Lee et al. [4] used their model to propose a modular method of approaching the problem that can deal with different recovery procedures from different airlines.

Even though there are dramatic advances in the field of operation research and computer science over the past decade, there are still lots of work to be done to come up with the efficient methodologies and software to solve the complicated real-life problems. Many of these problems are currently unsolvable, not because current computer systems are too slow or have too little memory, but simply because it is too difficult to determine what the computer program should do to solve these complicated problems. If the computer program could learn to solve the problems by itself, this would result in a great contribution in the field of operation research and computer science. RL is one such approach that makes the computer program to learn while trying to solve the complex decision-making problems. RL dates back to the early days of cybernetics and work in statistics, psychology, neuroscience, and computer science. In the last decade, it has rapidly attracted increasing interest in the machine learning and artificial intelligence communities. RL has significant potential in advancing parameters and policy optimization techniques. Sutton and Barto [5] and Bertsekas and Tsitsiklis [6] provide an excellent background reading for this field. Comprehensive literature surveys of pre-1996 research have been published by Kaelbling et al. [7] and Mahadevan [8]. Creighton and Nahavandi [9] developed a MATLAB toolbox to allow an RL agent to be rapidly tuned to optimize a multipart serial line. Aydin and Oztemel [10] successfully applied RL agents to dynamic job-shop scheduling problems. Other agent-based work in the job scheduling field has also been completed by Jeong [11], Zhang and Dietterich [12], Reidmiller and Reidmiller [13], and Schneider et al. [14]. Several research groups have recently focused on RL agent applications in manufacturing. Paternina-Arboleda and Das [15] used a SMART algorithm on a serial production line to optimize the preventative maintenance in a production inventory system. Mahadevan et al. [16] used this same algorithm and touched upon the integration of intelligent agents using RL algorithms with commercial DES packages. Mahadevan and Theocharous [17] also examined a manufacturing application using RL technique.

8.2.2 Literature Review for Service Call Center Staff Planning

Service call centers are the common way for many companies to communicate with their customers. In the customer point of view, the quality of service at the service call center usually reflects the operational efficiency of the company. Thus, the performance of the service call center is very essential for the survival of the company within our highly competitive service-driven economy. One important issue that many companies have to face is staff planning at their customer service call centers. At a service call center, hundreds of agents may have to answer to several thousands of telephone calls per hour. In addition, the number of calls is usually uncertain and is quite hard to predict from one time period to the next. The design of such an operation has to be based on solid scientific principles. Sze [18] discusses a queuing model of telephone operators at the Bell Communication Research Company, Inc. The queuing model is used to approximate the effects of several features such as general service times, abandonment, reattempts, etc. The results have proved to be quite useful in planning and managing the operator staffing for the service call center. Andrews and Parsons [19] have developed an economic-optimization model for telephone agent staffing at L. L. Bean. The model provides half an hour target staffing levels to an automated scheduler, which generates the specific on duty tours for each individual telemarketing operator. Chen and Henderson [20] discuss difficulties in using historical arrival rates to determine the staffing levels for a call center with priority customers. Fukunaga et al. [21] describe a staff scheduling system for contact centers called Blue Pumpkin Director. Borst et al. [22] use an *M*/*M*/*N* queuing model to build a model for staffing large service call centers with a large number of agents (*N*). Atlason et al. [23] use simulation and an iterative cutting plane method to find the staffing plan that minimizes the overall cost of a service system subject to a certain service level over multiple time periods. Atlason et al. [24] use simulation and analytic center cutting plane method to find the staffing plan that minimizes the overall staffing cost in an inbound call center subject to a certain service level. Deslauriers et al. [25] consider a blend call center with both inbound and outbound calls. They present a continuous time Markov chain models to solve the problem. Mourtada [26] considers the staffing problem at the Continental airline service call center and uses the RL technique to solve the problem.

8.3 Simulation-Based Optimization: Rl Technique

In our everyday life, we have to make many decisions. For each decision that we make, we can observe the immediate impact of that decision. It may not be a smart idea to use the immediate consequence of the decision as the only measurement for the quality of that decision. In fact, many decisions that we make have both the immediate consequence and the long-term consequences. By not properly accounting for the relationship between immediate and long-term consequences when making the important decisions, the resulting decisions may not have the good overall performance. For example, in a marathon racing, a racer who runs with the full speed at the beginning may be the leader in the initial phase of the race (good immediate consequence). Unfortunately, this may result in depleting the reserved energy very quickly and finally may result in a very poor finish (poor overall performance).

In this section, we first discuss the theoretical concepts and the general mathematical notations, formulations, and solution methodology of the sequential decision-making problems under uncertainty such that both immediate and long-term consequences have to be considered when making the decision. We will also discuss the difficulties in formulating and solving these models for the real-world decision-making problems. We then introduce the general concepts of RL, which properly combines simulation and optimization techniques to solve these complex decision-making problems under uncertainty.

8.3.1 Sequential Decision-Making System and Markov Decision Process

Figure 8.1 illustrates the general framework of the sequential decision-making system. At a particular point in time before making the decision, hereafter called *decision*

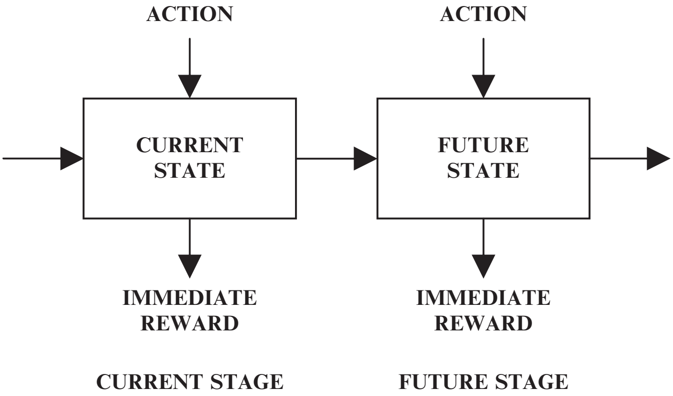


Fig. 8.1 General framework of sequential decision-making systems

stage, the decision maker has to carefully observe the information about the surrounding environment. This information will be hereafter called *system state*. Based on the system state information, the decision maker selects a possible decision, hereafter called *action*. After the appropriate action is chosen, decision maker receives the immediate consequence, hereafter called *immediate reward*, and the system stochastically evolves with some probability distributions, hereafter called *transition probability*, to a new system state at the next decision stage. At this decision stage, the decision maker again faces a similar decision-making problem.

Let us now define the general mathematical notations for the sequential decision-making problems. Let T denote the set of all possible decision stages. Let S denote the set of all possible system states. If at a particular decision stage, the decision maker observes that the system is in the state $s \in S$, he or she may select an action a from the set of all possible actions in the system state s, A_s. Let $A = \cup s \in sA_s$ denote the set of all possible actions. As the result of selecting an action $a \in A_s$ in the system state $s \in S$ at the decision stage $t \in T$, the decision maker receives an immediate reward of $r_t(s,a)$ and the system state at the next decision stage is determined by the transition probability $p_t(\cdot|s,a)$. In this section, we assume that the set S and A_s and the values of $r_t(s,a)$ and $p_t(\cdot|s,a)$ do not vary with different decision stages. Because of these assumptions, we will use the notations $r(s,a)$ and $p(\cdot\backslash s,a)$ instead of $r_t(s,a)$ and $p_t(\cdot|s,a)$ respectively for the rest of this chapter. We also assume that sets S and A_s are finite and the reward $r(s,a)$ is bounded for all system states and actions. The collection of objects $[T,S,A_s, p(\cdot|s,a), r(s,a)]$ is referred to as a *Markov decision process* (MDP). To formulate the mathematical models for sequential decision-making problems under uncertainty, decision makers have to properly define this collection of objects. The book written by Puterman [27] summarizes the detailed methodologies and theoretical concepts about MDP.

The solutions of the sequential decision-making problems under uncertainty are represented as *policies*. A *policy* normally refers to the set of selected actions for each state of the system. Without loss of generality, we assume that the decision makers are searching for the policy that maximizes the expected value of the overall reward of the system. Let v(s) denote the maximum expected value of the overall reward of the system when the system is initially in the system state *s*. Based on these notations, we can solve for the optimal policy for a given sequential decision-making problem by solving the following set of equations, hereafter called *optimality equations*:

$$v(s) = \max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j) \right\} \forall s \in S \tag{8.1}$$

where $\lambda \in (0,1)$ represents the discounting factor per each decision stage for the future rewards. If the optimality equations can be solved, the optimal policy for each system state *s* is

$$a^*(s) \in \arg\max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j) \right\} \tag{8.2}$$

Once all elements of MDP are identified and the optimality equations are constructed, we can apply the following algorithm called value iteration algorithm to find an ε-optimal policy and the approximated value of $v(s) \forall s \in S$.

8.3.1.1 Value Iteration Algorithm

Step 1: Select arbitrary real values for $v^0(s) \forall s \in S$, specify $\varepsilon > 0$, and set $n = 0$.

Step 2: For each $s \in S$, compute $v^{n+1}(s)$ by

$$v^{n+1}(s) = \max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v^n(j) \right\} \tag{8.3}$$

Step 3: If $\| \vec{V}^{n+1} - \vec{V}^n \| < \varepsilon (1-\lambda)/2\lambda$, go to step 4. Otherwise increase the value of n by 1 and return to step 2. Note that V^n is a vector of size $|S|$ containing $\vec{v}^n(s) \forall s \in S$ as its elements.

Step 4: For each $s \in S$, choose

$$a^*(s) \in \arg\max_{a \in A_s} \left\{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v^{n+1}(j) \right\} \forall s \in S \tag{8.4}$$

and stop.

After the algorithm terminates, the resulting values of $v^{n+1}(s)$ and $a^*(s) \; \forall s \in S$ represent the optimal expected values of the overall reward and the ε-optimal policy of the considered problem, respectively.

Unfortunately, formulating and solving the real-world decision-making problems as a MDP is not an easy task. In many cases, obtaining the complete information on $r(s,a)$ and $p(\cdot \mid s,a)$ is a very difficult and time-consuming process. This process may involve a number of complex mathematical terms consisting of the joint probability distribution of many random variables. Furthermore, many unrealistic assumptions may have to be made in the process of obtaining this information. This phenomenon is hereafter called the *curse of modeling* of the MDP. If we can solve the sequential decision-making problems with the efficient methodology that does not require the exact close-form formulation of $r(s,a)$ and $p(\cdot \mid s,a)$, this methodology would be really attractive and would really be applicable to solve many complex real-world problems. In fact, RL is one of the methodologies that have the promising potential to perform this task. In the following subsection, we will discuss the RL technique and how to apply the technique to solve the complex sequential decision-making problems under uncertainty.

8.3.2 RL Technique

Because MDP is seriously cursed by the curse of modeling for some real-world decision-making problems, the methodology such as RL, which does not require the close-form formulations of rewards and transition probabilities, is of our interest in this subsection. It is worth noting that unlike the solution obtained from MDP, which is guaranteed to be optimal, the resulting solution obtained from the RL may only be just suboptimal. RL nicely combines the simulation technique with the solution methodology of MDP and normally produces a high quality solution to the problem.

The key idea of RL is to approximately solve the optimality equations, which may not be represented in the close-form formulations by utilizing the simulation models. Let us introduce the notation $Q(s,a)\forall s\in S$, $\forall a\in A_s$ such that

$$\begin{aligned} Q(s,a) &= r(s,a) + \lambda \sum_{j\in S} p(j|s,a)v(j) \quad \forall s\in S, \forall a\in A_s \\ &= \sum_{j\in S} p(j|s,a)(r(s,a,j) + \lambda\, v(j)) \quad \forall s\in S, \forall a\in A_s \end{aligned} \tag{8.5}$$

where $r(s,a,j)$ represents the immediate reward by making the action a in the system state s and the next system state is j. By using this notation of $Q(s,a)$, the optimality equations can be rewritten as

$$v(s) = \arg\max_{a\in A_s} \{Q(s,a)\} \forall s\in S \tag{8.6}$$

$$a^*(s) = \arg\max_{a\in A_s} \{Q(s,a)\} \forall s\in S \tag{8.7}$$

These equations imply that if we can calculate the value of $Q(s,a)$ $\forall s\in S$, $\forall a\in A_s$, we can easily obtain the value of $v(s)$ and $a^*(s)$ $\forall s\in S$, which are the decided solutions of the problem. We will now concentrate on the methodology for approximating the value of $Q(s,a)$ $\forall s\in S$, $\forall a\in A_s$. By using the definition of $Q(s,a)$, we can obtain the following equations:

$$Q(s,a)=\sum_{j\in S} p(j|s,a)\left(r(s,a,j)+\lambda \max_{b\in A_s}\{Q(j,b)\}\right) \quad \forall s\in S,\forall a\in A_s$$

$$= E\left(r(s,a,j)+\lambda \max_{b\in A_j}\{Q(j,b)\}\right) \quad \forall s\in S,\forall a\in A_s \tag{8.8}$$

As this equation indicates, calculating the value of $Q(s,a)$ $\forall s\in S$, $\forall a\in A_s$ involves the expectation operation, which can be obtained by using the simulation model and the result from the following Robbins-Monro algorithm. The Robbins-Monro algorithm is the algorithm developed in 1951 by Robbins and Monro [28] for estimating the population mean of a random variable from the sample. Let X denote the considered random variable and let xi denote the value of the ith independent sample of X. Let X^n denote the value of the sample average of x_i from $i = 1$ to n. From the strong law of large number, we can obtain the following relationship between $E(X)$, X^n, and x_i:

$$E(X)=\lim_{n\to\infty}\left(\sum_{i=1}^{n} \mathrm{x}_i/n\right)$$

$$=\lim_{n\to\infty}\left(X^n\right) \tag{8.9}$$

The Robbins-Monro algorithm utilizes the relationship between X^n and X^{n+1} and suggests the iterative procedure for calculating the value of $E(X)$. The relationship between X^n and X^{n+1} can easily be derived as follows where $\alpha^n = 1/n$:

$$X^{n+1}=\sum_{i=1}^{n+1} x_i/(n+1) \tag{8.10}$$

$$X^{n+1}=\left(\sum_{i=1}^{n} X_i+X_{n+1}\right)/(n+1)$$

$$X^{n+1}=\left(nX^n+X_{n+1}\right)/(n+1)$$

$$X^{n+1}=\left(n/(n+1)\right)X^n+(1/(n+1))X_{n+1}$$

$$X^{n+1}=\left(1-\alpha^{n+1}\right)X^n+(\alpha^{n+1})X_{n+1} \tag{8.11}$$

By using this relationship, we can iteratively calculate the value of X^1, X^2, …, X^N after obtaining the sample information about the random variable X and can use the value of X^N as the approximation to $E(X)$ if N is a significantly large number. It is worth mentioning that the sample information of the random variable can be generated by using the simulation model and this is exactly the idea of RL. RL uses the basic idea of Robbins-Monro algorithm in calculating the expected value of the random variable to iteratively calculate the value of $Q(s,a)$ $\forall s\in S$, $\forall a\in A_s$ and finally obtain the values of $v(s)$ and $a^*(s)$ $\forall s\in S$. The algorithm iteratively calculates the value of $Q(s,a)\ \forall s\in S,\ \forall a\in A_s$ by generating a series of numbers $Q^1(s,a), Q^2(s,a), \ldots, Q^N(s,a)$ by utilizing the following relationship:

$$Q^{n+1}(s,a) = \left(1-\alpha^{n+1}\right)Q^n(s,a) + \left(\alpha^{n+1}\right)\left(r(s,a,j) + \lambda \max_{b\in A_j}\left\{Q^n(j,b)\right\}\right) \tag{8.12}$$

This calculation will be executed each time the action a is made in the system state s and the system evolves into the system state j. This relationship allows us to calculate the value of $Q(s,a)\ \forall s\in S,\ \forall a\in A_s$ without knowing the close-form formulation of rewards and transition probabilities because the value of $r(s,a,j)$ can be obtained from the simulation model. By utilizing this idea, the basic procedure of RL can be summarized as follows.

8.3.2.1 Basic RL Procedure for Discounted MDP

Step 1: Initialize the values of $Q(s,a) = 0$ and $N(s,a) = 0\ \forall s\in S,\ \forall a\in A_s$. Set $i = 0$ and N = maximum number of iterations (large integer number).
Step 2: Let s denote the current state of the system (from the simulation model). Randomly select an action from set A_s, each with equal probability. Let a denote the selected action.
Step 3: By selecting this action a in the system state s, the simulation model will be used to determine the next state of the system in the following decision stage. Let j denote this next system state. In addition, the simulation model will also be used to determine the value of $r(s,a,j)$

Set

$$N(s,a) \leftarrow N(s,a) + 1,\ i \leftarrow i + 1, \text{ and } \alpha = 1/N(s,a).$$

Step 4: Update the value of $Q(s,a)$ by using the following relationship.

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\left(r(s,a,j) + \lambda \max_{b\in A_j}\left\{Q(j,b)\right\}\right)$$

Step 5: If $i<N$, update the current system state $s = j$ and return back to step 2. Otherwise proceed to step 6.
Step 6: Calculate and return the following values of $v(s)$ and $a^*(s)\ \forall s\in S$:

$$v(s) = \max_{a\in A_s}\left\{Q(s,a)\right\} \quad \forall s\in S \quad \textit{and} \quad a^*(s) = \arg\max_{a\in A_s}\left\{Q(s,a)\right\} \quad \forall s\in S$$

Figure 8.2 illustrates the general framework of this RL algorithm.

Note that more sophisticated methods of selecting the action can be implemented to improve the overall performance of the algorithm. In this subsection, we only present the basic idea of the algorithm that randomly selects an action for each iteration. In the following section, we apply the RL technique to the staff planning

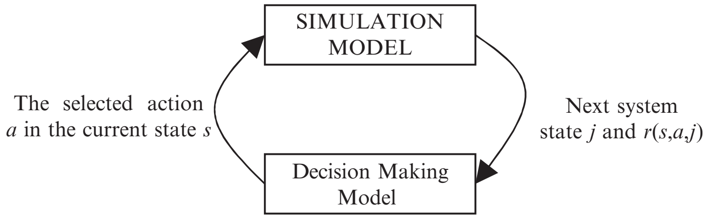


Fig. 8.2 General framework of the reinforcement learning algorithm

problem of the airline's service call center. All results illustrate the very promising potential of the algorithm to solve this complex real-world problem.

8.4 Case Study on Airline's Cargo Service Call Center Planning

In today's business, many companies are aggressively racing to improve their customer service to increase their customer satisfaction in order to survive in the current highly competitive business environment. The Airline industry is no exception. Airline companies are constantly looking for new and innovative ways to keep their customers satisfied and to stay in the market. To do so, airline companies must ensure a high level of customer service 24 h a day, 7 days a week. This requires hard work and dedication from their employees at every level. Although employees do not lack any dedication, it is the correct staffing policy that poses a challenge for the managers at the airline's service call center. Efficient staff planning could make all the difference between success and failure in managing the customer service call center. Staffing managers are facing the challenge of deciding on the number of customer service agents required for each month to properly answer incoming customer calls in order to meet the certain service level with the minimum overall cost possible.

In this section, we apply the RL technique to the staff planning problems by using the real data obtained from one of the largest airline companies in USA. One of this airline's service centers is the Cargo Service Center (CSC). The CSC provides cargo booking and tracking services. The CSC handles 10 different types of customer calls that are divided as follows: (1) international (general service calls); (2) animal; (3) elite; (4) mortuary; (5) globalink; (6) SAS; (7) service recovery; (8) JFK; (9) Spanish; and (10) AMS.

In this chapter, we will concentrate our attention only on four major types of calls at the CSC, namely international, animal, elite, and mortuary, which comprise over 90% of the overall number of calls. The objective of this work is to decide on the number of agents required for each month at each of the four different types of customer calls. It is necessary to mention that both international and animal calls at CSC are currently handled by the same group of agents. This means that the data for both international and animal calls can be consolidated to create one set of data for this study. The airline company would like to set the service level for these four different types of customer calls as follows. For animal and international calls, 80% of all calls should be answered within 20 s of their arrival. For elite calls, 80% of all calls should be answered within 20 s of their arrivals. For mortuary calls, 70% of all calls should be answered within 20 s of their arrivals. To meet these service-level requirements, the number of agents on duty must be carefully decided and allocated. To gain better understanding of the system, multiple observational visits are made to the CSC. Observations included listening to the four different types of calls and observing their processes. The managers of the CSC are also of great help for us to understand the overall system. After acquiring enough information about the overall system and its processes, the system is then translated into a high-level flowchart, which is eventually transformed into the detailed simulation model. As a call enters the system, it will be classified as animal, international (GS), elite, or mortuary call. The call will then be answered immediately if there is at least one available agent at the time of its arrival, or else it will wait in the split specific queue. Each call split has its own queue and its own 1–800 number. The airline company has a policy that if a call arrives in its specific queue and there are seven calls already waiting in that queue, then the call will be rolled over to a different available queue. This is to keep customers' wait times at minimum and ensure that all agents are properly utilized, since some call splits have lower volumes than the others. The call will then wait in the next queue, given that it has less than seven calls waiting in it already, until the next agent becomes available and the call will be answered. Finally, once the call has been answered, it exits the system. If all queues are full, the incoming calls will not be answered. Figure 8.3 illustrates the

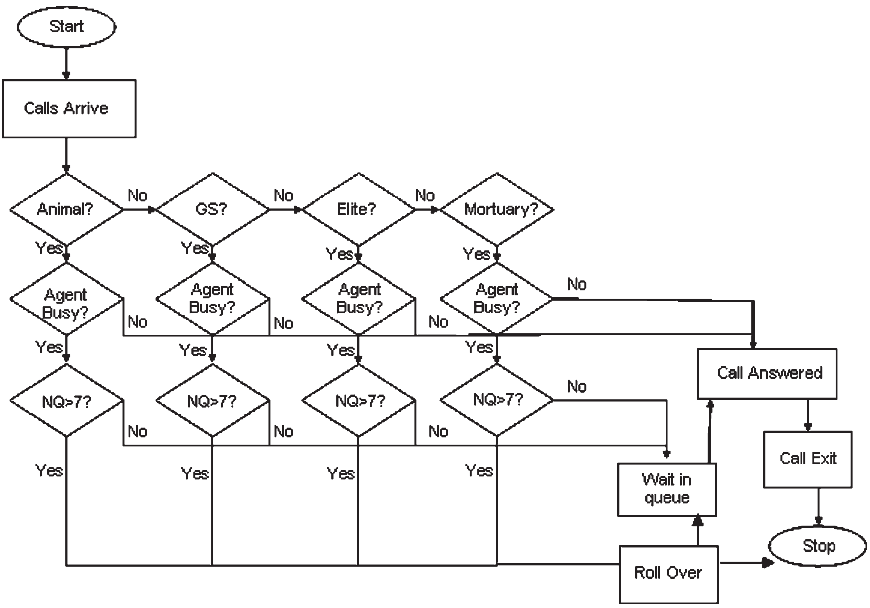


Fig. 8.3 Flowchart of customer calls routing at the CSC

flowchart of the customer call routing at the CSC where the notation NQ denotes the number of calls waiting in the queue.

8.4.1 Data Collection and Analysis for Constructing the Simulation Model

Accurate data analysis is the key fundamental in developing any simulation model. The performance of the simulation model can only be as good as the accuracy of the input data. With that in mind, the data collection and analysis is one of the most important tasks of this research. For this work, real data for an entire year are used to construct the simulation model of the service call center. The data in this study are obtained by using the historical information from the airline company. The airline company records these data for the different call splits and stores them in the company database. Note that the data used for this research are the year 2005 data. The data used in constructing the simulation model include (1) the interarrival time for each type of calls on each day of each month for the entire year and (2) the service time of each type of calls on each day of each month for the entire year.

Once the data had been collected and analyzed, appropriate probability distributions of these parameters are determined by utilizing the ARENA 10.0 input analyzer [29]. Input analyzer is a statistical analysis program included in the simulation software package called ARENA. This program takes a set of raw data as its input and generates a list of probability distributions that best fits the data. Figure 8.4 illustrates an example output of ARENA input analyzer. Once all required probability distributions of the model parameters are obtained, the detailed simulation model of the entire system is then developed by utilizing the simulation software package ARENA 10.0.

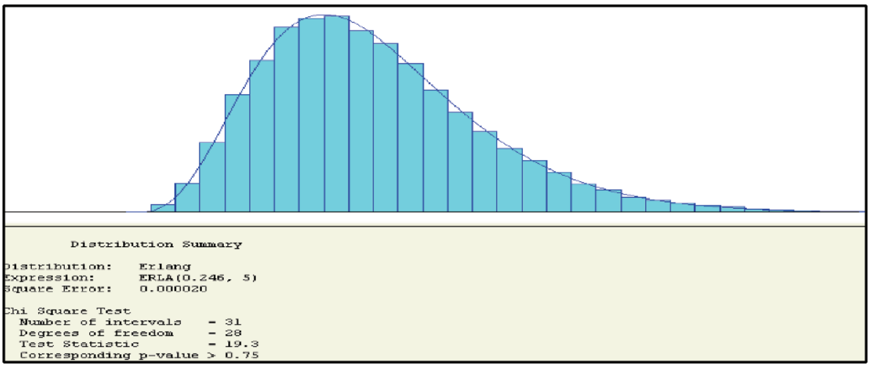


Fig. 8.4 An example output of ARENA input analyzer

8.4.2 RL Model for the Service Call Center Problems

After the simulation model of the service call center has been developed, some components of MDP have to be determined in order to implement the RL technique. These components are (1) the state space (S); (2) the action space for each possible system state s (A_s); (3) the reward structure; and (4) the decision stage (T). In this problem, the state information consists of the number of calls from the previous month and the current calendar month. For example, in the month of May, one of the possible states is s = (12,000 calls, May) if the number of calls in April was 12,000. After the system state information is observed, the possible action is basically the number of agents available to work in the current month. The reward structure of this problem is the numerical quantity that indicates how well a certain policy performs under certain circumstances. Deciding on the structure of the reward is somewhat challenging when modeling a service call center. The reward has to be measured in terms of the number of answered calls, the number of dropped calls, the number of calls with long queue waiting time, the hiring and firing costs, and the number of agents working at the service call center. In this model, the following formulation is used to calculate the reward value of making a certain action in a particular state.

Reward = [(profit per call) × (number of answered calls)] - [(monthly salary per agents) × (number of agents)] – [(penalty) × (number of calls that do not meet the required service level)] – [(Hiring cost per agent) × (the number of new agents)] - [(Firing cost per agent × the number of agents fired)]

This reward value can easily be obtained from the simulation model. Finally, the decision stage is the time period between each pair of the decision-making processes. In this work, the decision stage is the beginning of each month when the decision maker is required to decide on the number of working agents for each type of calls. Once all these components are identified, the RL technique is then applied to solve the considered decision problem. The simulation and decision-making models of the RL are executed on a Windows XP-based Pentium(R) 4 CPU 3.60 GHz personal computer with 4.00 GB RAM using Arena 10.0 and Visual Basic for Application (VBA) programming language. MS-Excel is used for the case study input and output database. Table 8.1 summarizes the recommended staffing policy for international or animal type of calls.

Table 8.1 Recommended staffing policy for the international or animal call split

Month (s)	No. of last month calls	No. of agents
February	Any value	31 agents
January, March, April, June, September	Any value	33 agents
May, July, August	Any value	35 agents
October, November, December	Any value	37 agents

Table 8.2 Recommended staffing policy for the elite call split

Month (s)	No. of last month calls	No. of agents
February, March, April, June, October, November, December	Any value	7 agents
January, August	Any value	8 agents
May, September	<4,700 calls	7 agents
May, September	≥4,700 calls	8 agents
July	<5,100 calls	8 agents
July	≥5,100 calls	9 agents

Table 8.3 Recommended staffing policy for the mortuary call split

Month (s)	No. of last month calls	No. of agents
April, May	All value	4 agents
June, August, September, October	All value	5 agents
January	<2,700	4 agents
January	≥2,700	5 agents
February	<2,400	4 agents
February	≥2,400	5 agents
March	<1,900	4 agents
March	≥1,900	5 agents
July	<2,250	4 agents
July	≥2,250	5 agents
November	<2,200	5 agents
November	≥2,200	6 agents
December	<2,450	5 agents
December	≥2,450	6 agents

Tables 8.2 and 8.3 illustrate the recommended staffing policies for the elite call split and the mortuary call split, respectively.

The current staffing policy at the CSC is to use 34.5, 7, and 8 full-time equivalents (FTEs) working on answering the international or animal, elite, and mortuary types of calls, respectively. An FTE consists of either a full-time employee or two part time employees. In the following subsection, we will compare the performance of these recommended solutions with the performance of the current policy used by the airline company. All results illustrate the improvements in the system performances resulting form the recommended solutions over the current policy.

8.4.3 Case Study Result and Performance Comparison

In this subsection, our goal is to statistically compare the performances of the policies recommended by the RL model and the performances of the current policy utilized by the airline company (original). To do so, another simulation model is developed to read a specific staffing policy as the input. This simulation model will then evaluate the input policy and will calculate a number of important performance measures of the system as the output. These results of each policy are analyzed and statistically compared. In this research, the following characteristics are used to measure the performance of the service call center: (1) the average number of calls that do not meet the required service level; (2) the average number of calls that are dropped; (3) the average utilization of agents; (4) the average waiting time in queue of each call; and (5) the overall cost per month of the system.

Based on the results obtained from 100 simulation years run, the values of these characteristics are calculated and recorded for each policy. After obtaining the values of these characteristics, statistical hypothesis testing procedures are performed in order to analyze and compare the performances of these two policies. These statistical hypotheses are summarized in the Tables 8.4 and 8.5. The mean values of these characteristics generated by the simulation model are compared between these two policies by utilizing the standard *t*-test. Note that the *t*-test is very robust for testing these hypotheses even if the data are not normally distributed when the sample sizes are large, which is the case for the examined data sets in this research.

If the null hypothesis contained in Table 8.4 is rejected for a specific performance measure, then we can conclude that the RL solution performs better in that characteristic. If the null hypothesis contained in Table 8.5 is rejected for a specific performance measure, we can conclude that the solution generated by the current plan performs better in that characteristic. If we fail to reject the hypotheses in both Tables 8.4 and 8.5 for a specific performance measure, we can conclude that there is no statistical difference between the two policies in that characteristic. Before applying the *t*-test to test these hypotheses, *F*-test is first used to check for the equality of variances between the two data sets: The null hypothesis (HO) of the F-test states that the variances of these two data sets are equal, while the alternative hypothesis (Ha) of the F-test states that the variances of these two data sets are different. The results from the *F*-test will determine the type of *t*-test to be used. The detailed information about statistical hypothesis testing with the *t*-test and the *F*-test can be studied in the book written by Johnson [*30*].

Table 8.4 The first set of statistical hypotheses for performance comparison

Characteristic	H_0	H_a
Mean number of bad calls	Means are equal	Original > RL
Mean number of dropped calls	Means are equal	Original > RL
Mean utilization of agents	Means are equal	RL > Original
Mean queue time	Means are equal	Original > RL
Mean monthly cost	Means are equal	Original > RL

Table 8.5 The second set of statistical hypotheses for performance comparison

Characteristic	H_0	H_a
Mean number of bad calls	Means are equal	Original < RL
Mean number of dropped calls	Means are equal	Original < RL
Mean utilization of agents	Means are equal	RL < Original
Mean queue time	Means are equal	Original < RL
Mean monthly cost	Means are equal	Original < RL

Table 8.6 Summary of the performance comparison for animal or GS call type

	Animal or GS type calls				
	No. of bad calls	Average queue time	No. of dropped calls	Average utilization	Average monthly cost
January	Δ	Δ	Δ	Δ	O
February	X	X	Δ	O	O
March	Δ	Δ	Δ	O	O
April	Δ	Δ	Δ	O	O
May	Δ	Δ	O	X	Δ
June	Δ	Δ	Δ	O	O
July	Δ	Δ	O	X	Δ
August	Δ	Δ	O	X	Δ
September	Δ	Δ	Δ	Δ	O
October	O	O	O	X	Δ
November	O	O	O	X	Δ
December	O	O	O	X	O
Overall	Δ	Δ	O	Δ	O

Table 8.7 Summary of the performance comparison for elite call type

	Elite type calls				
	No. of bad calls	Average queue time	No. of dropped calls	Average utilization	Average monthly cost
January	O	O	O	X	O
February	Δ	Δ	Δ	Δ	Δ
March	Δ	Δ	Δ	Δ	Δ
April	Δ	Δ	Δ	Δ	Δ
May	Δ	Δ	Δ	X	Δ
June	Δ	Δ	Δ	Δ	Δ
July	O	O	O	X	O
August	O	O	O	X	O
September	Δ	Δ	Δ	Δ	Δ
October	Δ	Δ	Δ	Δ	Δ
November	Δ	Δ	Δ	Δ	Δ
December	Δ	Δ	Δ	Δ	Δ
Overall	O	O	Δ	Δ	Δ

After performing the hypothesis testing procedures with the value of the type I error probability of 0.05, the results are obtained and summarized for each call type (animal or GS, elite, and mortuary). Tables 8.6–8.8 contain the summary information on the test results for animal or GS, elite, and mortuary call types, respectively, for each month and for the overall year. The following notations are used in these tables for ease in interpreting these results.

X: This notation indicates that the mean of the RL model was statistically significantly worse than the mean of the original model.

O: This notation indicates that the mean of the RL model was statistically significantly better than the mean of the original model.

Table 8.8 Summary of the performance comparison for mortuary call type

	Mortuary type calls				
	No. of bad calls	Average queue time	No. of dropped calls	Average utilization	Average monthly cost
January	O	D	X	O	D
February	Δ	Δ	Δ	O	Δ
March	O	Δ	Δ	O	Δ
April	Δ	Δ	Δ	O	Δ
May	O	Δ	Δ	O	Δ
June	Δ	Δ	Δ	Δ	Δ
July	O	Δ	X	O	Δ
August	Δ	Δ	Δ	Δ	Δ
September	Δ	Δ	Δ	Δ	Δ
October	Δ	Δ	Δ	Δ	Δ
November	O	O	O	X	Δ
December	O	O	O	X	Δ
Overall	O	O	Δ	O	Δ

Δ : This notation indicates that there is no statistically significant difference between the mean of the RL model and the mean of the original model.

Keeping in mind the results from the overall performance comparisons, we can come to the following conclusions. For the animal or GS call type, the staffing policy generated by the RL technique statistically outperforms the current staffing policy in the average number of dropped calls and the average monthly cost criteria. There are no statistically significant differences between the performances of these two policies for other criteria. For the elite call type, the staffing policy generated by the RL technique statistically outperforms the current staffing policy in the average number of bad calls and the average waiting time in queue criteria. There are no statistically significant differences between the performances of these two policies for other criteria. For the mortuary call type, the staffing policy generated by the RL technique statistically outperforms the current staffing policy in the average number of bad calls, the average waiting time in queue, and the average agent utilization criteria. There are no statistically significant differences between the performances of these two policies for other criteria.

8.5 Summary

Simulation and optimization are clearly two of the most powerful fields in the study of operation research and management science. Combining these two techniques together is definitely a promising concept for solving the real-world complex decision-making problems. In this chapter, the basic concepts of the simulation-based optimization technique, namely the RL, are explained and discussed in detail. We then apply the RL technique to determine the staffing policy for the airline service call center. Statistical hypothesis testing procedures are used to perform the performance comparisons between the recommended policy and the current policy. All results illustrate that the policy generated by the RL is superior to the current policy in a number of performance measures. This illustrates the promising potential of the simulation-based optimization techniques in generating the high quality solution for the complex decision-making problems in practice.

References

1. Gosavi, A., Simulation-Based Optimization: Parametric Optimization Techniques & Reinforcement Learning, Berlin Heidelberg New York, Springer, 2003.
2. Kleinman, N.L., Hill, S.D., and Ilenda, V.A., Simulation optimization of air traffic delay cost, Proceedings of the Winter Simulation Conference, Washington, DC, 1177–1181, 1998.
3. Rosenberger, J.M., Schafer, A.J., Goldsman, D., Johnson, E.L., Kleywegt, A.J., and Nemhauser, G.L., SIMAIR: A stochastic model of airline operations, Proceedings of the Winter Simulation Conference, 2000.
4. Lee, L.H., Hunag, H.C., Lee, C., Chew, E.P., Jaruphongsa, W., Yong, Y.Y., Liang, Z., Leong, C.H., Tan, Y.P., Namburi, K., Johnson, E., and Banks, J., Discrete event simulation model for airline operations: SIMAIR, Proceedings of the Winter Simulation Conference, 2003.
5. Sutton, R.S. and Barto, A.G., Reinforcement learning: An introduction, Cambridge, MA, MIT Press, 1998.
6. Bertsekas, D.P. and Tsitsiklis, J.N., Neuro-Dynamic Programming, Optimization and Neural Computer Series, 3, Belmont, MA, Athena Scientific, 1996.
7. Li Littman, M.L., Moore, A.W., and Kaelbling, L.P., Reinforcement Learning: A Survey, Brown University/Carnegie Mellon University, Providence, RI/Pittsburgh, PA.
8. Mahadevan, S., Average reward reinforcement learning: foundations, algorithms, and empirical results, Machine Learning, 22 (1), 159–195, 1996.
9. Creighton, D.C. and Nahavandi, S., Optimizing discrete event simulation models using a reinforcement learning agent, Winter Simulation Conference, Vol. 2, pp. 1945–1950, 2002.
10. Aydin, M.E., and Öztemel, E., Dynamic job shop scheduling using reinforcement learning agents, Robotics and Autonomous Systems, 33, 169–178, 2000.
11. Jeong, K., Conceptual frame for development of optimized simulation-based scheduling systems, Expert Systems with Applications, 18, 299–306, 2000.
12. Zhang, W. and Dietterich, T.G., A reinforcement learning approach to job-shop scheduling, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1114–1120, 1995.
13. Reidmiller, S. and Reidmiller, M., A neural reinforcement learning approach to learn local dispatching policies in production scheduling, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.
14. Schneider, J.G., Boyan, J.A., and Moore, A.W., Value function based production scheduling, Proceedings of the Fifteenth International Conference on Machine Learning, 24–27 July 1998.
15. Paternina-Arboleda, C.D. and Das, T.K., Intelligent dynamic control policies for serial production lines, IIE Transactions, 33, 65–77, 2001.
16. Mahadevan, S., Marchalleck, N., Das, T.K., and Gosavi A., Self-improving factory simulation using continuous-time average-reward reinforcement learning, Proceedings of the Fourteenth International Conference on Machine Learning, 202–210, 1997.
17. Mahadevan, S. and Theochaurus, G., Optimizing production manufacturing using reinforcement learning, Proceedings of the Eleventh International FLAIRS Conference, 372–377, 1998.

18. Sze, D.Y., A queuing model for telephone operator staffing, Operation Research, 32, 229–249, 1993.
19. Andrews, B. and Parsons, H., Establishing telephone-agent staffing levels through economic optimization, Interfaces, 23, 14–20, 1993.
20. Chen, B.P.K. and Henderson, S.G., Two issues in setting call centre staffing levels, Annals of Operations Research, 108, 175–192, 2001.
21. Fukunaga, A., Hamilton, E., Fama, J., Andre, D., Matan, O., and Nourbakhsh, I., Staff Scheduling for Inbound Call Centers and Customer Contact Centers, Blue Pumpkin Software, Sunnyvale, California, 2002.
22. Borst, S., Mandelbaum, A., and Reiman, M.I., Dimensioning large call centers, Operation Research, 52, 17–34, 2004.
23. Atlason, J., Epelman, M.A., and Henderson, S.G., Call center staffing with simulation and cutting plane methods, Annals of Operations Research, 127, 333–358, 2004.
24. Atlason, J., Epelman, M.A., and Henderson, S.G., Optimizing call center staffing using simulation and analytic center cutting plane methods, Department of Industrial and Operations Engineering, University of Michigan/School of Operations Research and Industrial Engineering, Cornell University, Ann Arbor, MI/Ithaca, NY, 2005.
25. Deslauriers, A., Ecuyer, P.L., Pichitlamken, J., Ingolfsson, A., and Avramidis, A.N., Markov chain models of a telephone call center with call blending, Computer and Operations Research, 34, 1616–1645, 2005.
26. Mourtada, I., Reinforcement Learning Model for Continental Airline's Service Center, M.S. Thesis, Department of Industrial Engineering, the University of Houston, May 2007.
27. Puterman, M., Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley Series in Probability and Mathematical Statistics, New York, Wiley, 1994.
28. Robbins, H. and Monro, S., A stochastic approximation method, Annals of Mathematical Statistics, 22, 400–407, 1951.
29. Kelton, W.D., Sadowski, R.P., and Sturrock, D.T., Simulation with Arena, 4th edition, New York, NY, Mc Graw Hill, 2007.
30. Johnson, R.A., Miller & Freund's Probability and Statistics for Engineers, 7th edition, Upper Saddle River, NJ, Pearson Prentice Hall, 2005.

Chapter 9
Robotics and Autonomous Robots

Henry I. Ibekwe[1] and Ali K. Kamrani[2]

Abstract Autonomous robots are rapidly changing manufacturing processes and industrial production systems. It is now an integral component essential for improved productivity and efficiency in industrial mechanized plants. In most modern large scale production facilities, tasks such as welding, forming, drilling, milling, and locating are completely performed by robots and the degree of autonomy will vary by industry. The emerging trend is to improve the robots' perceptual ability and cognition, thereby making them less reliant on an external controller and human intervention. This process is called autonomy, hence the name autonomous robots.

In this chapter, we shall discuss robot kinematics, basic task planning, and robot vision. Emphasis will be on the robot components that inherently improve robot autonomy. Arguably, the most important yet relatively underdeveloped robotic components are robot vision systems. The ability for a robot to "see" is an extremely difficult task because of the complex nature of visual perception. Comparable to human vision, an enormous amount of information is required to be processed and translated to the robot controllers. Humans perform visual tasks effortlessly without really understanding the tremendous processing ability of the brain. Till date, there are still major breakthroughs to be made in understanding vision and vision system as its application in robotics will significantly improve the autonomy of robots.

9.1 Introduction To Robotics

Robotics as defined by the *American Heritage Dictionary* is the science or study of the technology associated with the design, fabrication, theory, and application of robots. Robot, on the contrary, is a mechanical device that sometimes resembles a

[1]Department of Industrial Engineering, University of Houston, Houston, Texas 77004 USA

[2]Associate Professor, Department of Industrial Engineering, University of Huston, Houston, Texas 77004, USA

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

human and is capable of performing a variety of often complex human tasks on command or by being programmed in advance. According to Fu et al. [13], the Robot Institute of America defines industrial robots as "… a reprogrammable multifunctional manipulator designed to move materials, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks."

The study of robotics over the years has seen an immense growth in both technological application and otherwise. The recent advances in the manufacturing processes have necessitated the need to enable robots to be more autonomous. Autonomy simply means the ability of the robot to be independent, that is, intelligent. It should be understood that the mimicking of human intelligence and neural function is a relatively nascent research area and has significant strides to overcome in order to achieve this.

9.1.1 Application of Robots

There are numerous applications of robots today. They are applied anywhere from a highly sophisticated space robotic arm used by NASA's space exploration program to Lego® gadgets used for entertainment. Robots can also be found in various manufacturing industries, military applications, space exploration, and remote/hazardous environment. The application in industries has significantly increased in recent years because of improved productivity, reliability, and quality of end-products. This led to many manufacturing industries reformatting and remodeling plant layout to accommodate these changes.

9.1.1.1 Manufacturing Applications

The application of robots in the manufacturing industry includes operations such as drilling, cutting, deburring, parts installation, welding, painting, and inspecting. These tasks are just a few that can be performed at a high level of accuracy by the utilization of robots. Fewer system failures will occur because of improper machining and manufacturing than would otherwise have been performed by humans. An example of this occurs when uneven welds are created by machinists during the welding process. This can lead to failure of high-stress component within critical systems such as airplanes, bridges, and space shuttles. An example of a welding robot is shown in Fig. 9.1

Robots are vastly utilized in the electronics manufacturing because of the need for extreme accuracy, usually in the range of nanometers. Such levels of accuracy are nearly, if not, impossible to be performed by the average personnel. A pristine environment is also an added advantage. Tasks performed by the robots will minimize the possibility of debris and dust accumulation within the facility.

Painting and finishing are other common applications in the manufacturing environment. Automobile production lines utilize large number of robot manipulators for these operations. Uniformity and high accuracy is necessary during the paint and finishing phases.

Fig. 9.1 Fanuc Arc-Mate 120-i Robot Welding Cell (*Courtesy-Burns Machines*)

9.1.1.2 Assembly and Packaging Applications

Assembly operations have been quite cumbersome for application of robots; however, evolving techniques are reducing the level of difficulty required while maintaining quality standards. An example of this is a situation whereby difficulty arises when parts must be located and identified, carried in a particular order with many obstacles around the setup, fitted together, and then assembled [20]. Highly automated packaging companies solely apply robots for these purposes. The end result is often significant direct and indirect savings from reduced personnel cost to minimized product damage.

9.1.1.3 Remote and Hazardous Operations

Some tasks may involve high risk of fatality for humans. Robots that execute these tasks are often uniquely designed for very specific "workspace" conditions. Citing

Fig. 9.2 Mars rover (*Courtesy-NASA*)

the NASA Mars rover as an example, it was designed to collect vital research data about the inhabitable and rugged terrain of the planet Mars. Although there were minor glitches during the mission, the design of the autonomous rover (Fig. 9.2) has proven highly successful.

Remote-operated/autonomous robots have also found extensive use in underwater operations. Extremely high pressure at deep sea levels is an attribute to the dangers of deep sea exploration. Only recently could sunken naval, merchant, or cruise ships be explored. These robots can also be used for trans-Atlantic cable laying, deep sea dredging, and underwater pipeline inspection with the aid of vision systems.

9.1.1.4 Healthcare Applications

Imagine robots helping doctors perform complex operations like heart surgery or surgery for cancer using only a few tiny incisions. At first thought, it may come across as science fiction, but such tasks have been reliably performed tens of thousands of times already. This field of medicine is called *robotic-assisted minimally invasive surgery*. The design and production of a surgical robot is a formidable task since humans (a complex system) are now the "workpiece." Numerous variables have to be considered and a high level of redundancy must be implemented in the design of the robotic surgical system.

Minimal invasive surgery (MIS) is performed using narrow, long-shafted surgical instruments and cameras, which are inserted through small incisions (Figs. 9.3 and 9.4) that serve as ports of entry to the body (e.g., abdominal wall) to reach the surgical

Fig. 9.3 The *da Vinci*® surgical system patient cart (*Courtesy of Intuitive Surgical, Inc.*)

site [12]. For many patients, the potential advantages of having an operation with a robotic surgical system include significantly less pain, less blood loss and need for transfusions, less risk of infection, a shorter hospital stay, quicker recovery time, and better outcomes, in many cases. Shorter hospital stay and simplified postoperative care may also lower hospital overhead and improve overall operational efficiencies. Potential surgeon benefits may include improved visualization, which is provided by a 3-D high-definition camera; improved precision, dexterity, and control provided by miniaturized, wristed instrumentation, tremor filtration, and motion scaling; and an intuitive interface that provides optimized hand-eye alignment.

The most complex subsystem of a surgical robot is the end effector (the instrument tip). The end-effector design must mimic the dexterity (skill in using the hands) of a human hand and wrist. The desire to perform surgery through the smallest incisions possible limits the available surgical maneuvers possible, as well as the "degrees of freedom"-or the angles and direction in which the "wrist" can move [12].

Fig. 9.4 Robotic-assisted surgery setup (*Courtesy of Intuitive Surgical, Inc.*)

9.1.2 Classes of Robots

Robots are classified according to their coordinate frames. These coordinate frames define the motion capabilities of the robot and are important in the robot kinematics analysis. They are Cartesian coordinate, cylindrical coordinate, spherical coordinate, articulated coordinate, and the selective compliance assembly robot arm (SCARA) robots. A brief discussion of these coordinate systems will follow. Figure 9.5 identifies these robot classes.

9.1.2.1 Cartesian Coordinate Robots

Robots that fall under this class are often called rectangular or gantry robots and consist of three linear joints or axes. An example of a Cartesian coordinate robot is the IBM manufactured RS-1 robot. It has three prismatic joints and is modeled using the Cartesian coordinate system $(X, Y, Z) \rightarrow (X = a, Y = b, Z = c)$. Within a workspace, it travels linearly along any of the axis. During initialization it is important to note that starting axis and configuration settings are either pre-determined or dependent on various variables such as job type, work piece, and total repositioning time. Another similar robot is the *gantry robot* which operates on the same principle as the Cartesian coordinate robot. In this case, the robot is mounted on an overhead gantry.

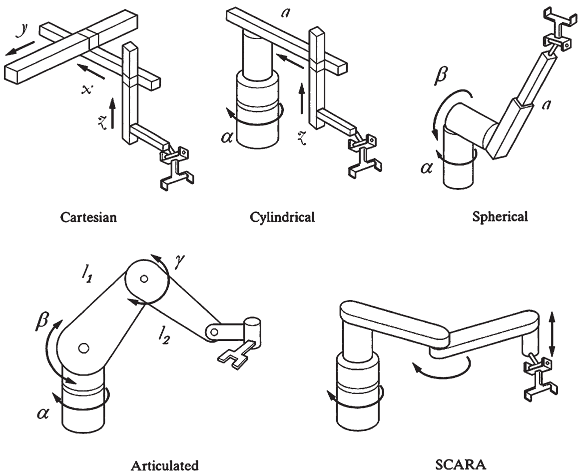


Fig. 9.5 Classes of robots (*Courtesy*-Niku [20])

9.1.2.2 Cylindrical Coordinate Robots

The cylindrical coordinate robots have two prismatic joints or linear axes and one revolute joint or rotary axis. The workspace for this class of robot resembles a cylinder as the name implies. The equivalent Cartesian coordinates can be found by the following equations

$$X = a\cos\alpha \tag{9.1}$$

$$Y = a\sin\alpha \tag{9.2}$$

$$Z = c \tag{9.3}$$

9.1.2.3 Spherical Coordinate Robots

The spherical coordinate robots have one prismatic or linear axis and two revolute joints or rotary axes. They have significant application in the manufacturing industry such as welding, cutting, and material handling. The corresponding *X*, *Y*, *Z* coordinates are

$$X = a\cos\alpha\cos\beta \tag{9.4}$$

$$Y = a\sin\alpha\cos\beta \tag{9.5}$$

$$Z = c\sin\beta \tag{9.6}$$

9.1.2.4 Articulated Coordinate Robots

The articulated coordinate robots are sometimes referred to as revolute or anthropomorphic robots. In this robot class, all joints are revolute or rotary and are comparable to the human arm. They are also highly utilized in industries and provide improved flexibility from the rectangular, cylindrical, or spherical coordinate robots. These axes connect three rigid links and the base. The corresponding *X*, *Y*, *Z* locations are

$$X = [l_1 \cos \beta + l_2 \cos (\beta + \gamma)]\cos \alpha \qquad (9.7)$$

$$Y = [l_1 \cos \beta + l_2 \cos (\beta + \gamma)]\sin \alpha \qquad (9.8)$$

$$Z = l_1 \cos \beta + l_2 \cos (\beta + \gamma) \qquad (9.9)$$

9.1.2.5 SCARA Robots

This is the fifth class of robots. SCARA is the acronym for Selective Compliance Assembly Robot Arm. SCARA robots have two parallel rotary or revolute joints and a linear or prismatic joint. The two rotary joints enable the robot to relocate with the horizontal plane, while the linear joint enables it to relocate or move within the vertical plane. They have relatively better flexibility along the *x*-axis and the *y*-axis in comparison to that along the *z*-axis and hence are employed in assembly tasks.

9.1.3 Components of Robots

The robotic manipulator is composed of several main subsystems. They are the main frame or manipulator, actuators, sensors, controllers, end effector, and computer processor/operating systems. These subsystems interact in unison to produce the desired task to be performed by the robot.

9.1.3.1 Body/Main Frame

The main frame of a robotic manipulator is the rigid structure of the robot. They are serially connected to various joint configurations as discussed in Sect. 9.3. The frame can sometimes be called the manipulator. Care must be taken in using the term manipulator as the manipulator alone does not constitute a robot.

9.1.3.2 Actuators

Actuators are essentially the “driver” of a robot. They provide the necessary forces and moment to translate the robot from one spatial coordinate to another. There are numerous actuators that can be used in the design of a robotic system. The more commonly used are pneumatic cylinder systems, hydraulic cylinder systems, servomotors, etc. Obviously, the actuators will require some level of control. This process is performed via the use of a controller.

9.1.3.3 Sensors

Robot sensors enable the robots to have some level of intelligence. The sophistication and complexity of the sensors provide increased autonomy for the robot. They are linked with robot controllers, discussed in subsequent sections, that provide a feedback system where information gathered can be analyzed and processed. Real-time information of the robot's internal state is collected through these sensors and allows the robot to interact with its environment with relative flexibility. The types of sensors commonly used are tactile sensors, vision sensors, speech sensors, etc.

9.1.3.4 Controllers

The controller moderates the movement and motions of the links of the robot. It verifies each link and the proper spatial coordinates in order to perform a task; therefore, it acquires data from the main computer process and transfers the information to the actuators.

9.1.3.5 End Effector

The end effector is the last element in the linkage of the robot arm. It may be referred to as the most important part of the robot. Specific design tasks will dictate the type of end effector on a robot manipulator. While design tasks of a robot vary, the main body of the robot can be consistent and require little alteration. For example, if a robot that primarily performs a pick-and-place task need to be redesigned to perform welding operations, the end effector will simply have to be modified and very little change made to the main body.

9.1.3.6 Computer Processor/Operating Systems

The computer processor is the "brain" of the robot manipulator. It controls the decision making and receives input signals from the sensors. The computer processor transmits the output signals to the robot controller. These signals are fed to actuators, which, in turn, provide the necessary forces to drive the robot.

Operating system is the software component of the computer processing system that enables the programmer to write computer algorithms and codes meant to run the robot manipulator.

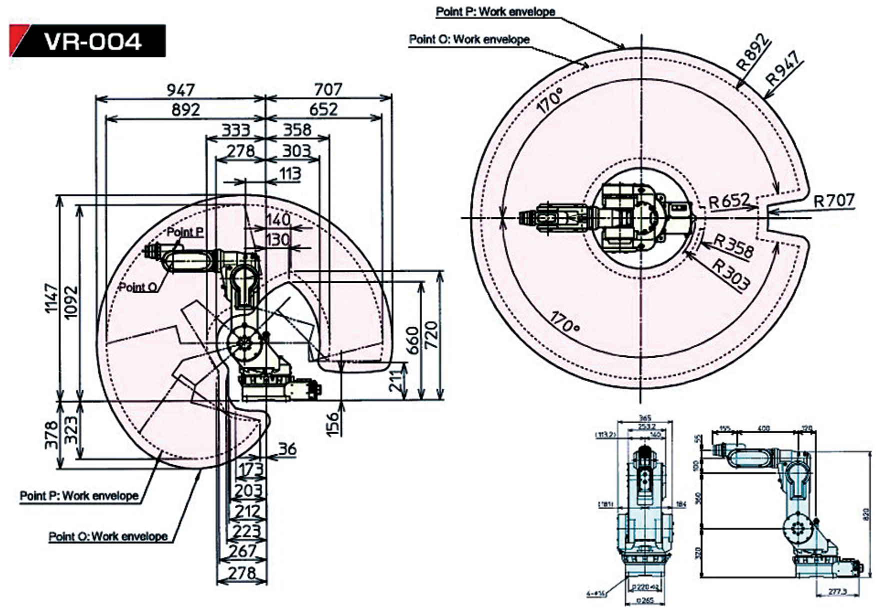


Fig. 9.6 Robot workspace (*Courtesy-US Dept of Labor*)

9.1.3.7 Robot Workspace

The robot workspace is the space within which the robot can manipulate to perform specified tasks. It is determined by the physical configuration of the robot, the size of the body, and the limits of the joint movements and therefore restricted to its work envelope (Fig. 9.6)

The definition of the workspace is extremely critical in the kinematics analysis, plant design, plant layout, and production planning of robots. To analyze forward kinematics, a set of equations that are related to a specific configuration of the robot will have to be developed. This is done by substituting the joint and link variables in these equations developed. Consequently, we may calculate the position and orientation of the robot. These equations will then be used to derive the inverse kinematics.

9.2 Robot Kinematics

A robot manipulator generally consists of a frame, a wrist, and an end effector. The frame is composed of the arm, shoulder, and elbow and all are connected by rigid links. They are designed to operate within a design workspace and are typically implemented in industries. An industrial robot is a general-purpose, computer-controlled manipulator consisting of several rigid links connected in series by revolute or prismatic joints [13]. It can be modeled as an open-loop articulated chain consisting of the aforementioned rigid links typically connected in series prismatic or revolute joints. These joints are powered by actuators which are devices that produce forces to drive the systems to specified positions and orientations.

Robot manipulator kinematics involves the mathematical analysis of the geometry of motion of a robot manipulator taking into consideration a fixed reference coordinate system. The forces, moments that generate these motions, will not be considered during analysis but rather reflects on the special displacement of the robot as a function of time. A strong background in elementary vector analysis will be a definite requisite force for the study and understanding of robot kinematics.

Kinematics pertaining to robots is divided into two groups, forward kinematics and reverse kinematics. If all positional/joint variables of all joints and links of the robot are known, then the robot end-effector (hand) position can be determined by forward kinematics. Conversely, if the positional/joint variables of the end effector are known, then all the positional and joint variables of the joints and links can be determined by inverse or reverse kinematics.

Arguably the most important component of a robot manipulator is the end effector since it will primarily perform the required task. Therefore, it is a necessity to identify the position of each joint variable space with respect to the position of the end effector. The flexibility of the end effector is also important in analysis as this will determine its possible orientation.

9.2.1 Representation of Translating Bodies

In order to adequately derive and describe robot kinematics models, we shall revise the matrix representation of a point and a vector in 3-D space. This will give better insight in understanding the translation of body-attached coordinate. The basic knowledge of matrix algebra and vector analysis is also a necessity.

9.2.1.1 Representation of a Point and a Vector in 3-D Space

Consider a point $\mathbf{p}_{xyz}$ in space (Fig. 9.7). We can represent this point by its three coordinates with respect to the reference coordinate frame.[1] That is,

$$\mathbf{p}_{xyz} = p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k} \tag{9.10}$$

[1] The reference coordinate frame is based on the *right-handed* Cartesian coordinate system and is called *orthonormal*.

where p_x, p_y, and p_z are the components of **p***xyz* along the *x*-axis, *y*-axis, and *z*-axis, respectively, and **i**, **j**, **k** are unit vectors along the *x*-axis, *y*-axis, *z*-axis, respectively. If there is a vector that originates from the origin (0, 0, 0) of the reference coordinate axis, then the vector from origin *O* to point *pxyz* can be represented as **p** such that

$$\mathbf{p}_{xyz} = p_x\mathbf{i} + p_y\mathbf{j} + p_z\,\mathbf{k} \tag{9.11}$$

Equation 9.11 shall subsequently be used to develop matrix transformation equations.

9.2.1.2 Rotation Matrices

Rotations matrices of the robot links having dimensions of a square matrix, $m \times m$ where $m = 3$, can be defined as a transformation matrix which functions on a position vector in a 3-D space. There are two coordinate systems, the reference coordinate system *OXYZ* and the body-attached coordinate system *OUVW* (Fig. 9.7). The components of the reference coordinate system, *OXYZ*, are *OX*, *OY*, and *OZ,* while the components of the body-attached coordinate system, *OUVW*, are *OU*, *OV*, and *OW*. The body-attached coordinate systems are fixed onto links of a robot and can translate or rotate.

Let $\mathbf{i}_x$, $\mathbf{j}_y$, and $\mathbf{k}_z$ be the unit vectors along the coordinate axes of *OXYZ* and let $\mathbf{i}_u$, $\mathbf{j}_v$, and $\mathbf{k}_w$ be the unit vectors along the coordinate axes of *OUVW.* Then a point **p** at rest and fixed with respect to the *OUVW* coordinate frame in space can be represented by its coordinates with respect to *OXYZ* and *OUVW.* Therefore

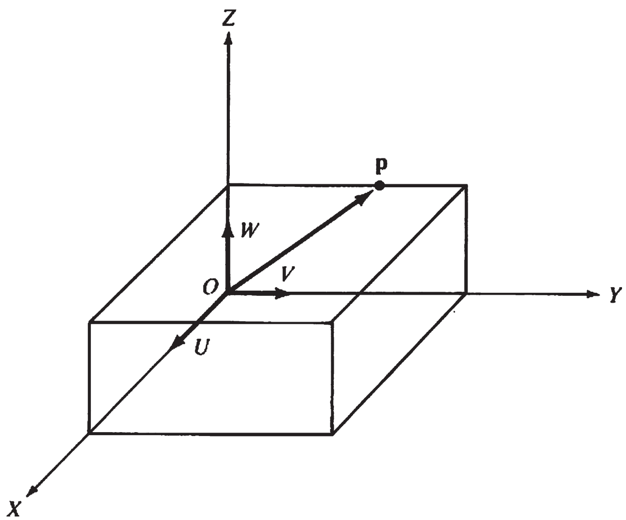


Fig. 9.7 Reference and the body-attached coordinate system

$$\mathbf{p}_{xyz} = (p_x, p_y, p_z)^T \tag{9.12}$$

$$\mathbf{p}_{uvw} = p_u, p_v, p_w \tag{9.13}$$

where $\mathbf{p}_{xyz} = \mathbf{p}_{uvw}$ meaning that they are at the same point in space but are based on different coordinate systems and are represented by a 3 × 1 matrix. If the coordinate reference frame *OUVW* is rotated, the problem is to develop a 3 × 3 transformation matrix **R** that will transform the coordinates of $\mathbf{p}_{uvw}$ to the coordinates expressed with respect to the reference coordinate frame *OXYZ*. In other words

$$\mathbf{P}_{xyz} = \mathbf{R}\ \mathbf{p}_{uvw} \tag{9.14}$$

This transformation will result in an *orthogonal* transformation. Rewriting **p***uvw* and **p***xyz* in terms of their components we have

$$\mathbf{p}_{uvw} = p_u\mathbf{i}_u + p_v\mathbf{j}_v + p_w\mathbf{k}_w \tag{9.15}$$

$$\mathbf{p}_{xyz} = p_x\mathbf{i}_x + p_y\mathbf{j}_y + p_z\mathbf{k}_z \tag{9.16}$$

where p_u, p_v, and p_w represent the components of **p** along *OU*, *OV*, and *OW* and p_x, p_y, and p_z represent the components of **p** along *OX*, *OY*, and *OZ*. Expressing **p** in terms of the components p_x, p_y, and p_z, we have

$$p_x=\mathbf{i}_x\bullet\mathbf{p}=\mathbf{i}_x\bullet\mathbf{i}_u\, p_u+\mathbf{i}_x\bullet\mathbf{j}_v\, p_v+\mathbf{i}_x\bullet\mathbf{k}_w\, p_w \tag{9.17}$$

$$p_y=\mathbf{j}_y\bullet\mathbf{p}=\mathbf{j}_y\bullet\mathbf{i}_u\, p_u+\mathbf{j}_y\bullet\mathbf{j}_v p_v+\mathbf{j}_y\bullet\mathbf{k}_w\, p_w \tag{9.18}$$

$$p_z=\mathbf{k}_z\bullet \mathrm{p}=\mathbf{k}_z\bullet\mathbf{i}_u\, p_u+\mathbf{k}_z\bullet\mathbf{j}_v \mathrm{p}_v+\mathbf{k}_z\bullet\mathbf{k}_w\, p_w \tag{9.19}$$

Rewriting in matrix form, we have

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_x\bullet\mathbf{i}_u & \mathbf{i}_x\bullet\mathbf{j}_v & \mathbf{i}_x\bullet\mathbf{k}_w \\ \mathbf{j}_y\bullet\mathbf{i}_u & \mathbf{j}_y\bullet\mathbf{j}_v & \mathbf{j}_y\bullet\mathbf{k}_w \\ \mathbf{k}_z\bullet\mathbf{i}_u & \mathbf{k}_z\bullet\mathbf{j}_v & \mathbf{k}_z\bullet\mathbf{k}_w \end{bmatrix} \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} \tag{9.20}$$

Let $\mathbf{R} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$, then:

$$\mathrm{R}=\begin{bmatrix} \mathbf{i}_x\bullet\mathbf{i}_u & \mathbf{i}_x\bullet\mathbf{j}_v & \mathbf{i}_x\bullet\mathbf{k}_w \\ \mathbf{j}_y\bullet\mathbf{i}_u & \mathbf{j}_y\bullet\mathbf{j}_v & \mathbf{j}_y\bullet\mathbf{k}_w \\ \mathbf{k}_z\bullet\mathbf{i}_u & \mathbf{k}_z\bullet\mathrm{j}_v & \mathbf{k}_z\bullet\mathbf{k}_w \end{bmatrix} \tag{9.21}$$

We can also develop a matrix **Q** that will transform the coordinates of $\mathbf{p}_{xyz}$ to the coordinates expressed with respect to the reference coordinate frame *OUVW*. That is

$$\mathbf{p}_{UVW}=\mathbf{Q}\mathbf{p}_{xyz} \tag{9.22}$$

Expressing **p** in terms of the components p_u, p_v, and p_w, we have

$$p_u=\mathbf{i}_u\cdot\mathbf{p}=\mathbf{i}_u\cdot\mathbf{i}_x p_x+\mathbf{i}_u\cdot\mathbf{j}_y p_x+\mathbf{i}_u\cdot\mathbf{k}_z p_x \tag{9.23}$$

$$p_v=\mathbf{j}_v\cdot\mathbf{p}=\mathbf{j}_v\cdot\mathbf{i}_x p_y+\mathbf{j}_v\cdot\mathbf{j}_y p_y+\mathbf{j}_v\cdot\mathbf{k}_z p_y \tag{9.24}$$

$$p_w=\mathbf{k}_w\cdot\mathbf{p}=\mathbf{k}_w\cdot\mathbf{i}_x \mathrm{p}_z+\mathbf{k}_w\cdot\mathbf{j}_y \mathrm{p}_z+\mathbf{k}_w\cdot\mathbf{k}_z p_z \tag{9.25}$$

Rewriting in matrix form, we have,:

$$\begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} = \begin{bmatrix} \mathbf{i}_u\bullet\mathbf{i}_x & \mathbf{i}_u\bullet\mathbf{j}_y & \mathbf{i}_u\bullet\mathbf{k}_z \\ \mathbf{j}_v\bullet\mathbf{i}_x & \mathbf{j}_v\bullet\mathbf{j}_y & \mathbf{j}_v\bullet\mathbf{k}_z \\ \mathbf{k}_w\bullet\mathbf{i}_x & \mathbf{k}_w\bullet\mathrm{j}_y & \mathbf{k}_w\bullet\mathbf{k}_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{9.26}$$

Let $\mathbf{Q}=\begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix}$, then

$$Q = \begin{bmatrix} \mathbf{i}_u\bullet\mathbf{i}_x & \mathbf{i}_u\bullet\mathbf{j}_y & \mathbf{i}_u\bullet\mathbf{k}_z \\ \mathbf{j}_v\bullet\mathbf{i}_x & \mathbf{j}_v\bullet\mathbf{j}_y & \mathbf{j}_v\bullet\mathbf{k}_z \\ \mathbf{k}_w\bullet\mathbf{i}_x & \mathbf{k}_w\bullet\mathrm{j}_y & \mathbf{k}_w\bullet\mathbf{k}_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{9.27}$$

Since $\mathbf{p}_{xyz} = \mathbf{R}\ \mathbf{p}_{uvw}$ and $\mathbf{p}_{uvw} = \mathbf{Q}\ \mathbf{p}_{xyz}$, we have

$$\mathbf{Q}=\mathbf{R}^{\mathrm{T}}=\mathbf{R}^{-1} \tag{9.28}$$

$$\therefore \mathbf{QR}=\mathbf{R}^{\mathrm{T}}\mathbf{R}=\mathbf{R}^{-1}\mathbf{R}=\mathrm{I}_3 \tag{9.29}$$

where $\mathbf{I}_3$ is a 3 × 3 identity matrix, that is, $\mathrm{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and since the vectors in the dot product are all unit vectors, it is also called *orthonormal* transformation. The problem now is to find the rotation matrix **R** such that if the *OUVW* coordinate system is rotated about any axis, it can be represented with respect to the *OXYZ*. We shall consider the three cases that will arise.

- Rotation of the *OUVW* system about the *OX*-axis by an angle α
- Rotation of the *OUVW* system about the *OY*-axis by an angle ϕ
- Rotation of the *OUVW* system about the *OZ*-axis by an angle θ

9.2.1.3 Rotation About the *OX*-Axis

For *case 1* (Rotation of the *OUVW* system about the *OX*-axis by an angle α) (Fig. 9.8), the rotation matrix $\mathbf{R}_{x,\alpha}$ can be derived as follows

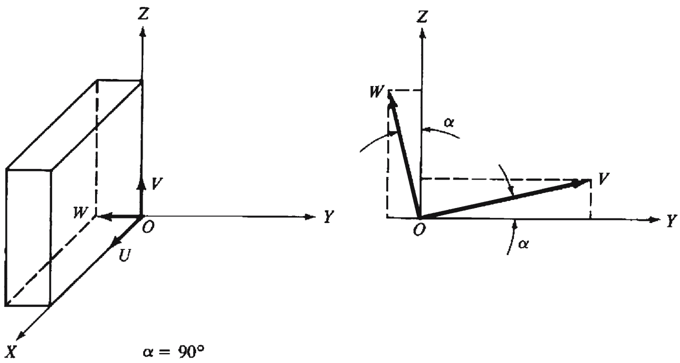


Fig. 9.8 Rotation of the *OUVW* system about the *OX*-axis

$$\mathbf{p}_{xyz} = \mathbf{R}_{x,\alpha}\, \mathbf{p}_{uvw} \tag{9.30}$$

$$\mathrm{R}_{x,\alpha} = \begin{bmatrix} \mathbf{i}_x \bullet \mathbf{i}_u & \mathbf{i}_x \bullet \mathbf{j}_v & \mathbf{i}_x \bullet \mathbf{k}_w \\ \mathbf{j}_y \bullet \mathbf{i}_u & \mathbf{j}_y \bullet \mathbf{j}_v & \mathbf{j}_y \bullet \mathbf{k}_w \\ \mathbf{k}_z \bullet \mathbf{i}_u & \mathbf{k}_z \bullet \mathbf{j}_v & \mathbf{k}_z \bullet \mathbf{k}_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{C}\alpha & -\mathrm{S}\alpha \\ 0 & \mathrm{S}\alpha & \mathrm{C}\alpha \end{bmatrix}$$

where $\mathbf{i}_x$ $\mathbf{i}_u$ and by the definition, the dot product of two vectors **a** and **b** will result in a scalar $\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos\theta$. Also to simplify matrix element notation we can write $\cos\alpha$ as $\mathrm{C}\alpha$ and similarly sinα as $\mathrm{S}\alpha$. This notation will be frequently used in further discussion.

9.2.1.4 Rotation About the *OY*-Axis

For *case 2* (Rotation of the *OUVW* system about the *OY*-axis by an angle φ) (Fig. 9.9), the rotation matrix $\mathbf{R}y,\varphi$ can be derived as follows

$$\mathbf{p}_{\mathrm{xyz}} = \mathbf{R}_{\mathrm{y},\phi}\, \mathbf{p}_{\mathrm{uvw}} \tag{9.31}$$

$$\mathrm{R}_{y,\phi} = \begin{bmatrix} \mathbf{i}_x \bullet \mathbf{i}_u & \mathbf{i}_x \bullet \mathbf{j}_v & \mathbf{i}_x \bullet \mathbf{k}_w \\ \mathbf{j}_y \bullet \mathbf{i}_u & \mathbf{j}_y \bullet \mathbf{j}_v & \mathbf{j}_y \bullet \mathbf{k}_w \\ \mathbf{k}_z \bullet \mathbf{i}_u & \mathbf{k}_z \bullet \mathbf{j}_v & \mathbf{k}_z \bullet \mathbf{k}_w \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} = \begin{bmatrix} \mathrm{C}\phi & 0 & \mathrm{S}\phi \\ 0 & 1 & 0 \\ -\mathrm{S}\phi & 0 & \mathrm{C}\phi \end{bmatrix}$$

9.2.1.5 Rotation About the *OZ*-Axis

For *case 3* (Rotation of the *OUVW* system about the *OZ*-axis by an angle θ) (Fig. 9.10), the rotation $\mathbf{R}_{z,\theta}$ can be derived as follows

$$\mathbf{p}_{xyz} = \mathbf{R}_{z,\theta}\, \mathbf{p}_{uvw} \tag{9.32}$$

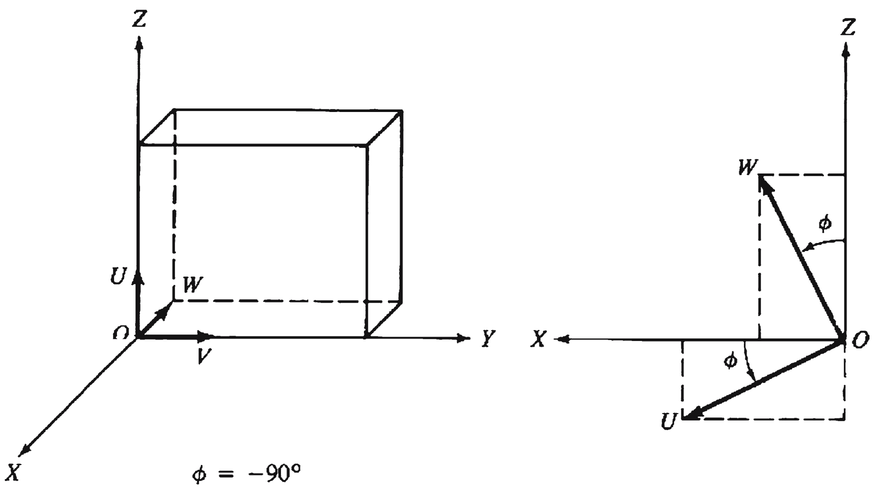


Fig. 9.9 Rotation of the *OUVW* system about the *OY*-axis

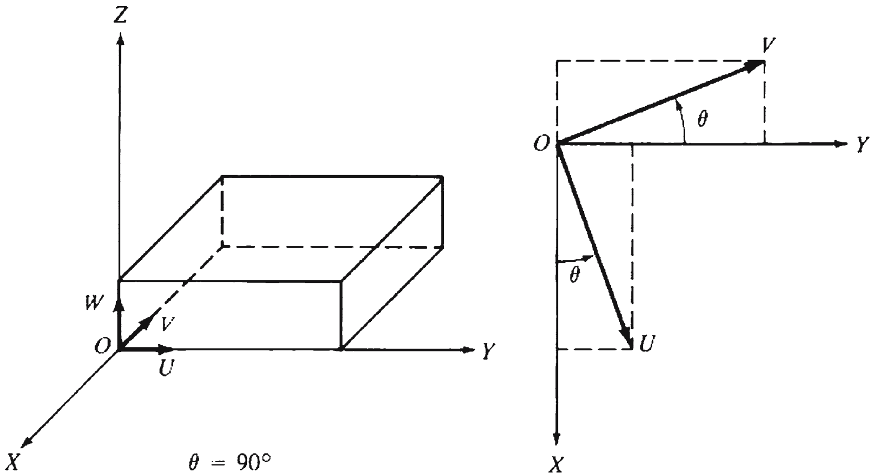


Fig. 9.10 Rotation of the *OUVW* system about the *OZ*-axis

$$
R_{z,y,\theta} = \begin{bmatrix} \mathbf{i}_x \bullet \mathbf{i}_u & \mathbf{i}_x \bullet \mathbf{j}_v & \mathbf{i}_x \bullet \mathbf{k}_w \\ \mathbf{j}_y \bullet \mathbf{i}_u & \mathbf{j}_y \bullet \mathbf{j}_v & \mathbf{j}_y \bullet \mathbf{k}_w \\ \mathbf{k}_z \bullet \mathbf{i}_u & \mathbf{k}_z \bullet \mathbf{j}_v & \mathbf{k}_z \bullet \mathbf{i}_w \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

Example 9.1

With respect to the *OUVW* coordinate system, a point **q** in 3-D space is given to be $\mathbf{q}_{uvw} = (3, 6, 4)^T$. Find the equivalent points with respect to the *OXYZ* coordinate system, if there has been (1) a 30° rotation about the *OX*-axis, (2) a 120° rotation about the *OY*-axis, and (3) a 45° rotation about the *OZ*-axis (*hint: there will be three different rotation matrices*).

Solution:

(1) With a 30° rotation about the *OX*-axis, the equivalent point is $\mathbf{q}_{xyz} = \mathbf{R}_{x,30°}\, \mathbf{q}_{uvw}$ and using the matrix form of Eq. 9.30, where $\alpha = 30°$,

$$R_{x,30^\circ} = \begin{bmatrix} \mathbf{i}_X \bullet \mathbf{i}_U & \mathbf{i}_X \bullet \mathbf{j}_V & \mathbf{i}_X \bullet \mathbf{k}_W \\ \mathbf{j}_Y \bullet \mathbf{i}_u & \mathbf{j}_Y \bullet \mathbf{j}_v & \mathbf{j}_Y \bullet \mathbf{k}_W \\ \mathbf{k}_Z \bullet \mathbf{i}_u & \mathbf{k}_Z \bullet \mathrm{j}_v & \mathbf{k}_Z \bullet \mathbf{i}_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 30^\circ & -\sin 30^\circ \\ 0 & \sin 30^\circ & \cos 30^\circ \end{bmatrix}$$

$$R_{x,30^\circ} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.8660 & -0.5 \\ 0 & 0.5 & 0.8660 \end{bmatrix}$$

$$\therefore q_{xyz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.8660 & -0.5 \\ 0 & 0.5 & 0.8660 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 3.1962 \\ 6.4641 \end{bmatrix}$$

(2) With a 120° rotation about the *OY*-axis, the equivalent point is $\mathbf{q}_{xyz} = \mathbf{R}_{y,120°}\mathbf{q}_{uvw}$ and using the matrix form of Eq. 9.31, where $\phi = 120°$,

$$R_{y,120^\circ} = \begin{bmatrix} \mathbf{i}_X \bullet \mathbf{i}_U & \mathbf{i}_X \bullet \mathbf{j}_V & \mathbf{i}_X \bullet \mathbf{k}_W \\ \mathbf{j}_Y \bullet \mathbf{i}_U & \mathbf{j}_Y \bullet \mathbf{j}_V & \mathbf{j}_Y \bullet \mathbf{k}_W \\ \mathbf{k}_Z \bullet \mathbf{i}_U & \mathbf{k}_Z \bullet \mathrm{j}_V & \mathbf{k}_Z \bullet \mathbf{k}_W \end{bmatrix} = \begin{bmatrix} \cos 120^\circ & 0 & \sin 120^\circ \\ 0 & 1 & 0 \\ -\sin 120^\circ & 0 & \cos 120^\circ \end{bmatrix}$$

$$R_{y,120^\circ} = \begin{bmatrix} -0.5 & 0 & 0.8660 \\ 0 & 1 & 0 \\ -0.8660 & 0 & -0.5 \end{bmatrix}$$

$$\therefore q_{xyz} = \begin{bmatrix} -0.5 & 0 & 0.8660 \\ 0 & 1 & 0 \\ -0.8660 & 0 & -0.5 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 1.9641 \\ 6 \\ -4.5981 \end{bmatrix}$$

(c)With a 45° rotation about the *OZ*-axis, the equivalent point is $q_{xyz} = R_{z,45^o}\, q_{uvw}$ and using the matrix form of Eq. 9.32, where $\theta = 45°$,

$$R_{Z,45^\circ} = \begin{bmatrix} \mathbf{i}_X \bullet \mathbf{i}_U & \mathbf{i}_X \bullet \mathbf{j}_V & \mathbf{i}_X \bullet \mathbf{k}_W \\ \mathbf{j}_Y \bullet \mathbf{i}_U & \mathbf{j}_Y \bullet \mathbf{j}_V & \mathbf{j}_Y \bullet \mathbf{k}_W \\ \mathbf{k}_Z \bullet \mathbf{i}_U & \mathbf{k}_Z \bullet \mathrm{j}_V & \mathbf{k}_Z \bullet \mathbf{k}_W \end{bmatrix} = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{z,45^\circ} = \begin{bmatrix} 0.7071 & -0.7071 & 0 \\ 0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore q_{xyz} = \begin{bmatrix} 0.7071 & -0.7071 & 0 \\ 0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} -2.1213 \\ 6.3639 \\ 4 \end{bmatrix}$$

9.2.1.6 Combination of the Rotation Matrices

The rotation matrices formulated in the previous sections are based on rotation about a single axis. A rotation matrix can be developed to include all rotations of the *OUVW* body-attached frame about the *OX*-axis, *OY*-axis, and *OZ*-axis, simultaneously. The *order* or *sequence* of rotation about each axis is of critical importance. In other words, a rotation of 30° about the *OX*-axis then 60° about the *OY*-axis and 45° about the *OZ*-axis is *not* equivalent to a rotation of 60° about the *OY*-axis then a 30° about the *OX*-axis and a 45° about the *OZ*-axis. It is also possible for the body-attached reference frame *OUVW* to rotate about its own axis. It is then necessary to understand that since both coordinate systems are coincident initially, the resulting rotation matrix is a 3 × 3 identity matrix. Second, if the *OUVW* system is rotated about one of the principal axes of the *OXYZ* system, then *premultiply* the resultant rotation matrix with a corresponding rotation matrix. Likewise, if the *OUVW* system is rotated about any of its principal axes, postmultiply the resultant matrix with a corresponding rotation matrix.

Consider the following rotation sequences:

α° about *OX*-axis → ϕ° about *OY*-axis → θ° about *OZ*-axis.

We may generate the rotation matrix for each case as follows:

$$R = R_{z,\theta} R_{y,\phi} R_{x,\alpha} \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix}$$

$$R = \begin{bmatrix} C\phi C\theta & S\alpha S\phi C\theta - C\alpha S\theta & C\alpha S\phi C\theta + S\alpha S\theta \\ C\phi S\theta & C\alpha C\theta + S\alpha S\phi S\theta & C\alpha S\phi S\theta + S\alpha C\theta \\ -S\phi & S\alpha S\phi & C\alpha C\phi \end{bmatrix}$$

Example 9.2

Once again referring to *Example 9.1*, formulate the rotation matrix if the given rotations of the *OUVW* system were in sequence, that is, a 30° rotation about the *OX*-axis, then a 120° rotation about the *OY*-axis, and finally a 45° rotation about the OZ-axis. What is the resulting location of the point $\mathbf{q}_{uvw} = (3, 6, 4)^T$ with respect to the *OXYZ*-axis after the sequence of rotation (*hint: there is only one rotation matrix computed*).

Solution:

$$\mathrm{R} = R_{z,\theta} R_{y,\phi} R_{x,\alpha} \begin{bmatrix} \mathrm{C}\theta & -\mathrm{S}\theta & 0 \\ \mathrm{S}\theta & \mathrm{C}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}\phi & 0 & \mathrm{S}\phi \\ 0 & 1 & 0 \\ -\mathrm{S}\phi & 0 & \mathrm{C}\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{C}\alpha & -S\alpha \\ 0 & \mathrm{S}\alpha & \mathrm{C}\alpha \end{bmatrix}$$

Since $\alpha = 30°$, $\phi = 120°$, and $\theta = 45°$, we have

$$\mathrm{R}_{uvw} = \mathrm{R}_{z,45}{\circ}\mathrm{R}_{y,120}{\circ}\mathrm{R}_{x,30}{\circ}$$

$$= \begin{bmatrix} \mathrm{C}45° & -\mathrm{S}45° & 0 \\ \mathrm{S}45° & \mathrm{C}45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}120 & 0 & \mathrm{S}120 \\ 0 & 1 & 0 \\ -\mathrm{S}120° & 0 & \mathrm{C}120° \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{C}30° & -S30° \\ 0 & \mathrm{S}30° & \mathrm{C}30° \end{bmatrix}$$

$$R_{uvw} = \begin{bmatrix} \mathrm{C}120°\mathrm{C}45° & \mathrm{S}30°\mathrm{S}120°\mathrm{C}45° - \mathrm{C}30°\mathrm{S}45° & \mathrm{C}30°\mathrm{S}120°\mathrm{C}45° + \mathrm{S}30°\mathrm{S}45° \\ \mathrm{C}120°\mathrm{S}45° & \mathrm{C}30°\mathrm{C}45° + \mathrm{S}30°\mathrm{S}120°\mathrm{S}45° & \mathrm{C}30°\mathrm{S}120°\mathrm{S}45° + \mathrm{S}30°\mathrm{C}45° \\ -\mathrm{S}120° & \mathrm{S}30°\mathrm{S}120° & \mathrm{C}30°\mathrm{C}120° \end{bmatrix}$$

$$R_{uvw} = \begin{bmatrix} -0.3536 & -0.3062 & 0.8839 \\ -0.3536 & 0.9186 & 0.1768 \\ -0.8660 & -0.25 & -0.4330 \end{bmatrix}$$

Therefore, the point $\mathbf{q}_{uvw}$ with respect to the *OXYZ*-axis after the sequence of rotation can be found by Eq. 9.14, where

$$\mathbf{q}_{xyz} = \mathbf{R}_{uvw}\mathbf{q}_{uvw} = \begin{bmatrix} -0.3536 & -0.3062 & 0.8839 \\ -0.3536 & 0.9186 & 0.1768 \\ -0.8660 & -0.25 & -0.4330 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 0.6378 \\ 5.1578 \\ -5.8301 \end{bmatrix}$$

(*Note that multiplying the rotation matrix result in Example 1.1 (3), (2), and (1), respectively, will result in the rotation matrix above.)*

9.3 Homogenous Representation

The representation of an n-component position vector by an $(n + 1)$-component vector is called homogeneous coordinate representation [13]. The transformation of an n-dimensional vector is executed in the $(n + 1)$-dimensional space and the physical n-dimensional is thus obtained by dividing the homogeneous coordinates by the $(n + 1)$th coordinate, u. Given a position vector $\mathbf{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ in 3-D space, the homogenous coordinate representation can be depicted by an augmented vector

$\mathbf{p} = (up_x, up_y, up_z, u)^T$. The fourth component of the homogenous matrix can then be considered a scaling factor or multiplier. When $u = 1$,

$$p = (p_x/u, p_y/u, p_z/u, 1)^T = (p_x, p_y, p_z, u)^T \tag{9.33}$$

The transformed homogenous coordinates of a position vector are the same as the physical coordinates of the vector. It is most useful in practice to let $u = 1$ where the homogeneous representation can be used equivalently as the physical coordinates vector. This will aid in fully defining all the possible transformations that can occur to a position vector. The homogeneous transformation matrix, **T** (also known as the T-matrix), can be expressed as a composition of four submatrices

$$T = \begin{bmatrix} \text{Rotation}_{(3\times3)} & \text{Positional}_{(3\times1)} \\ \text{Perspective}_{(1\times3)} & \text{Scaling}_{(1\times1)} \end{bmatrix} \tag{9.34}$$

The composition of these four submatrices can be expressed as a 4 × 4 matrix. The following are the 4 × 4 representation of the submatrices:

- ***Basic Homogeneous Rotational Matrices:***

$$T_{R(x,\alpha)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C & -S & 0 \\ 0 & S & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{T}_{R(y,\phi)} = \begin{bmatrix} C & 0 & S & 0 \\ 0 & 1 & 0 & 0 \\ -S & 0 & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_{R(z,\theta)} = \begin{bmatrix} C & -S & 0 & 0 \\ S & C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.35}$$

(*Note:* $C = \cos\theta$ and $S = \sin\theta$.)

- ***Basic Homogeneous Translation Matrix:***

$$\mathbf{T}_{\text{trans}} = \begin{bmatrix} 1 & 0 & 0 & \Delta X \\ 0 & 1 & 0 & \Delta Y \\ 0 & 0 & 1 & \Delta Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.36}$$

Another transformation will be necessary to define scaling. Scaling of a coordinates system is the proportion that the transformed coordinate system has to the original coordinate system (reference coordinate system). Scaling is performed at each axis. If we define the scaling factor along each axis as s_x, s_y, and s_z, then the transformation matrix is

$$\mathbf{p}' = \mathbf{s}^{\mathrm{T}}\mathbf{p} = [\mathrm{s}_x \mathrm{s}_y \mathrm{s}_z]\ \mathbf{p} \tag{9.37}$$

where **s** is the scaling vector for the *x*, *y*, *z* components of the original coordinate system. The 4 × 4 matrix representation of the scaling factor known as the basic homogenous scaling matrix is

$$\mathbf{T}_{\mathrm{s}} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.38}$$

Therefore, the appropriate combination of any of the 4 × 4 homogenous transformation matrices maps a vector denoted in the homogenous coordinates in relation to the *OUVW* coordinate system (body-attached frame) to the reference coordinate system *OXYZ* and can be given by

$$\mathbf{T} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.39}$$

9.4 Forward or Direct Kinematics

Assuming that the length of the robot links and the joint angles are all known, then the position of the end effector with respect to the reference coordinate system can be computed using forward kinematics analysis. Vector and matrix algebra are utilized to develop a systematic and generalized approach to describe and represent the location of the link of a robot arm with respect to a fixed reference frame. Since the links of a robot arm are allowed to rotate and/or translate with respect to a reference coordinate frame, a body-attached coordinate frame will be established along the joint axis for each link. This will significantly simplify computation.

To analyze forward kinematics, vector analysis and transformation must be discussed. The end effector can be displaced within its workspace through rotations and translations of the links with respect to the reference coordinate frame. We shall also significantly minimize computations of link transformations by utilizing matrix algebra. In 1955, the duo of Denavit and Hartenberg[2] proposed a method by using matrix algebra to describe and represent the spatial geometry of the links of a robot arm with respect to a fixed reference frame.

[2]The Denavit and Hartenberg research and proposals were published in the 1955 edition of the American Society of Mechanical Engineers (ASME) *Journal of Applied Mechanics*. The famous technique is widely used in robotics and referred to as the D-H model.

The direct kinematics problem is reduced to finding a transformation matrix that relates the body-attached coordinate frame, that is, the *OUVW* coordinate system, to the reference coordinate frame, in this case the *OXYZ* coordinate system. A 3 × 3 rotation matrix is used to describe the rotational operations of the body-attached frame with respect to the reference frame. The homogenous coordinates are used to represent position vectors in a 3-D space, and the rotation matrices are expanded to 4 × 4 homogenous transformation matrices to include the translational operations of the body-attached coordinate frame in order to position and orient the rigid body in space.

If we consider an orthonormal coordinate system composed of unit vectors ($\mathbf{x}i$, $\mathbf{y}i$, $\mathbf{z}i$), then we can use this system to denote the base coordinate frame, all links, and its corresponding joint axis, where $i = 1, 2, \ldots n$ and n represents the number of degrees of freedom. When the joint i is set in motion via an actuator, link i is displaced in accordance to the motions of link $i - 1$. Given that the ith coordinate system is fixed in link i, it is displaced along with the link i. Note that a rotary joint has one degree of freedom and every ($\mathbf{x}i$, $\mathbf{y}i$, $\mathbf{z}i$) coordinate frame of a robot arm corresponds to joint $i + 1$ and is stationary in relation to link i. Also the nth coordinate frame will be displaced along with the nth link, which incidentally is the end effector. We have previously labeled the 0th link as the base link and hence the 0th coordinate frame is then the base coordinate frame denoted as ($\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{z}_0$).

According to the three rules given by Fu et al. [13], every coordinate frame is determined and established on the basis of the following:

1. The $\mathbf{z}_{i-1}$ axis lies along the axis of motion of the ith joint.
2. The $\mathbf{x}_i$ axis is normal to the $\mathbf{z}_{i-1}$ axis, and pointing away from it.
3. The $\mathbf{y}_i$ axis completes the right-handed coordinate system as required.

From these rules, the location of the base coordinate frame 0 can be placed at any region within the base so far $\mathbf{z}_0$ axis lies along the axis of motion of the first joint (see Fig. 9.11).

The D-H representation of a rigid link depends on four geometric parameters associated with each link. These four parameters completely describe any revolute or prismatic joint [13]. They are defined as follows:

θ_i = the joint angle from $\mathbf{x}_{i-1}$ axis to the x_i axis about the $\mathbf{z}_{i-1}$ axis (using the right-hand rule)

d_i = the distance from the origin of the $(i-1)$th coordinate frame to the intersection of the $\mathbf{z}_{i-1}$ axis with the $\mathbf{x}_i$ axis along the $\mathbf{z}_{i-1}$ axis

a_i = the offset distance from the intersection of the $\mathbf{z}_{i-1}$ axis with the $\mathbf{x}_i$ axis to the origin of the ith frame along the x_i axis (or the shortest distance between the $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ axes)

α_i = the offset angle from the $\mathbf{z}_{i-1}$ axis to the $\mathbf{z}_i$ axis about the $\mathbf{x}_i$ axis (using the right-hand rule)

- In the case of a rotary joint, d_i, a_i, and α_i are the joint parameters which will be constant for a robot while θ_i is the joint variable that changes when link i is displaced or rotated in relation to the link $i - 1$.

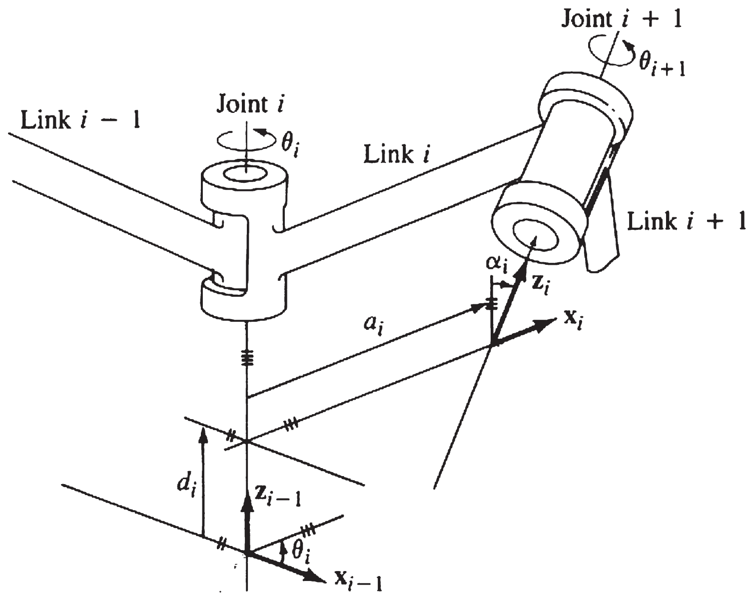


Fig. 9.11 Link coordinate system and its parameters (*Courtesy-Fu et al.* [13])

- In the case of a prismatic joint, θ_i, a_i, and α_i are the joint parameters and remain constant for a robot, while d_i is the joint variable.

Upon determining the D-H coordinate system for each link, a homogenous transformation matrix can easily be developed relating the ith coordinate frame to the $(i - 1)$th coordinate frame. Consequently, the point $\mathbf{r}_i$ denoted in the ith coordinate system can be denoted in the $(i - 1)$th coordinate system as $\mathbf{r}_{i-1}$. This can be achieved by the following (refer to Fig. 9.12):

- Rotate about the $\mathbf{z}_{i-1}$ axis an angle of θ to align the $\mathbf{x}_{i-1}$ axis with the $\mathbf{x}_i$ axis. The $\mathbf{x}_{i-1}$ axis is parallel to $\mathbf{x}_i$ and directed to the same direction.
- Translate along $\mathbf{z}_{i-1}$ axis a distance of d_i to bring the $\mathbf{x}_{i-1}$ and x_i axes into coincidence.
- Translate along the $\mathbf{x}_i$ axis a distance of a_i to bring the two origins as well as the x-axis into coincidence.
- Rotate about the $\mathbf{x}_i$ axis an angle of α_i to bring the two coordinate systems into coincidence.

We can thus represent each of these four processes by a basic homogenous rotation-translation matrix. Multiplying these matrices will result in a composite homogenous transformation matrix, ${}^{i-1}\mathbf{A}_i$, known as the D-H transformation matrix for adjacent coordinate frames, i and $i - 1$. We then have

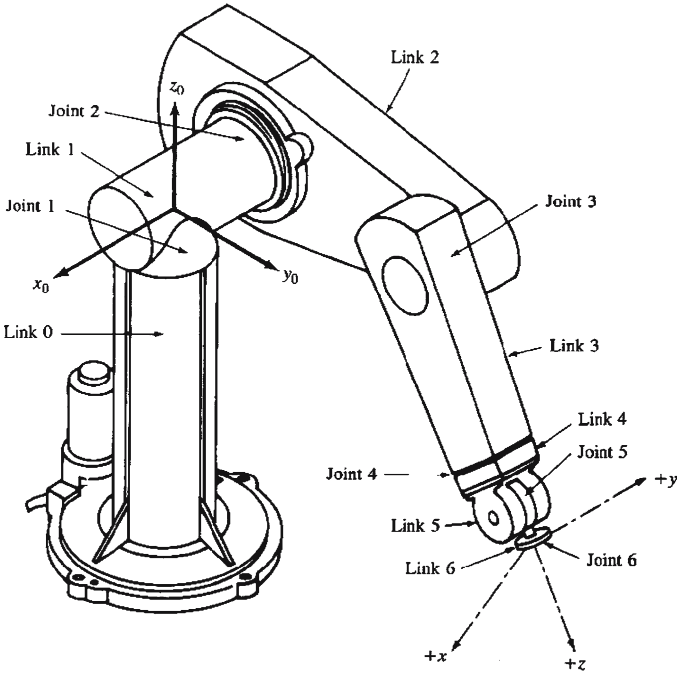


Fig. 9.12 A PUMA robot arm with joints and links (*Courtesy-Fu et al.* [13])

$$ {}^{\mathrm{i}-1}\mathbf{A}_{\mathrm{i}} = \mathbf{T}_{z,d} \cdot \mathbf{T}_{z,\theta} \cdot \mathbf{T}_{x,a} \cdot \mathbf{T}_{x,\alpha} \tag{9.40} $$

$$ {}^{i-1}\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ & \sin\alpha_i & \cos\alpha_i & \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i-1}\mathbf{A}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \sin\theta_i & -\sin\alpha_i \sin\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.41} $$

If we consider a revolute joint where α_i, a_i, and d_i are constant and θ_i is the joint variable for a revolute joint, then the inverse of the above transformation matrix can be found to be

$$\left[{}^{i-1}\mathbf{A}_i\right]^{-1} = {}^{i}\mathbf{A}_{i-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -a_i \\ -\cos\alpha_i \sin\theta_i & \cos\alpha_i \sin\theta_i & \sin\alpha_i & -d_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & -\sin\alpha_i \cos\theta_i & \cos\alpha_i & -d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.42}$$

Likewise for a prismatic joint, the joint variable is d_i, while α_i, a_i, and θ_i are the constants.

$${}^{i-1}\mathbf{A}_i = \mathbf{T}_{z,\theta} \bullet \mathbf{T}_{z,d} \bullet \mathbf{T}_{x,\alpha} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & 0 \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.43}$$

The inverse matrix for the prismatic joint is

$$\left[{}^{i-1}\mathbf{A}_i\right]^{-1} = {}^{i}\mathbf{A}_{i-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & 0 \\ \cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & \sin\alpha_i & d_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & -\sin\alpha_i \cos\theta_i & \cos\alpha_i & -d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.44}$$

From the $[{}^{i-1}A_i]^{-1}$matrix we can relate a point $\mathbf{p}i$ at rest in link i, and expressed in homogeneous coordinates with respect to the coordinate system i, to the coordinate system $i - 1$ established at link $i - 1$ by

$$\mathbf{p}_{i-1} = {}^{i-1}\mathbf{A}_i \mathbf{p}_i \tag{9.45}$$

where $\mathbf{p}_{i-1} = (\mathbf{x}_{i-1}, \mathbf{y}_{i-1}, \mathbf{z}_{i-1}, 1)^T$ and $\mathbf{p}_i = (\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i, 1)^T$

9.5 Reverse of Indirect Kinematics

In inverse kinematics analysis, if the position of the end effector with respect to the reference coordinate system is known, then the robot link parameters, variables, and joint angles can be calculated. It is the inverse of direct kinematics and can be solved using proven methods, namely, inverse transform, decoupling technique, [AU1] inverse transform technique, screw algebra, dual matrices, dual quanterion, and a host of other techniques. We shall concentrate on the *Inverse Transform Technique,* which was developed by Peiper in 1968.

If we have the transformation matrix ${}^{0}\mathbf{A}_6$ indicating the global position and the orientation of the end effector of a six degree of freedom robot in the base frame 0, then we can proceed to determining the joint variables and links positions. Assume that the geometry and individual transformation matrices ${}^0\mathbf{A}_1(q_1)$, ${}^1\mathbf{A}_2(q_2)$, ${}^2\mathbf{A}_3(q_3)$, ${}^3\mathbf{A}_4(q_4)$,${}^4\mathbf{A}_5(q_5)$, and${}^5\mathbf{A}_6(q_6)$ are given as the joint variables. Note that the inverse kinematics is attempting to determine the elements of vector **q** when a transformation is given as a function of the joint variables $q_1, q_2, q_3, \ldots q_n$, then we have, $0\mathbf{T}_n = 0\mathbf{A}_1(q_1)\ 1\mathbf{A}_2(q_2)\ 2\mathbf{A}_3(q_3)\ 3\mathbf{A}_4(q_4) \ldots n{-}1\mathbf{A}_n(q_n)$. Therefore, the problem is to find what the required values of joint variables are to reach a desired point in a desired orientation. The articulated arm transformation matrix derived from forward kinematics for a six degree of freedom robot is given as [*15*]

$$\mathrm{T}_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0\mathrm{A}_1{}^1\mathrm{A}_2{}^2\mathrm{A}_3{}^3\mathrm{A}_4{}^4\mathrm{A}_5{}^5\mathrm{A}_6 \tag{9.46}$$

The inverse kinematics problem is given as the following:

$$\begin{aligned}
{}^1\mathbf{T}_6 &= {}^0\mathbf{A}_1^{-1}\,{}^0\mathbf{A}_6 \\
{}^2\mathbf{T}_6 &= {}^1\mathbf{A}_2^{-1}\,{}^0\mathbf{A}_1^{-1}\,{}^0\mathbf{A}_6 \\
{}^3\mathbf{T}_6 &= {}^2\mathbf{A}_3^{-1}\,{}^1\mathbf{A}_2^{-1}\,{}^0\mathbf{A}_1^{-1}\,{}^0\mathbf{A}_6 \\
{}^4\mathbf{T}_6 &= {}^3\mathbf{A}_4^{-1}\,{}^2\mathbf{A}_3^{-1}\,{}^1\mathbf{A}_2^{-1}\,{}^0\mathbf{A}_1^{-1}\,{}^0\mathbf{A}_6 \\
{}^5\mathbf{T}_6 &= {}^4\mathbf{A}_5^{-1}\,{}^3\mathbf{A}_4^{-1}\,{}^2\mathbf{A}_3^{-1}\,{}^1\mathbf{A}_2^{-1}\,{}^0\mathbf{A}_1^{-1}\,{}^0\mathbf{A}_6 \\
\mathbf{I} &= {}^5\mathbf{A}_6^{-1}\,{}^4\mathbf{A}_5^{-1}\,{}^3\mathbf{A}_4^{-1}\,{}^2\mathbf{A}_3^{-1}\,{}^1\mathbf{A}_2^{-1}\,{}^0\mathbf{A}_2^{-1}\,{}^0\mathbf{A}_6
\end{aligned} \tag{9.47}$$

9.6 Robot Vision

Robot vision may be defined as the process of extracting, characterizing, and interpreting information from images of a 3-D world. This process is also known as computer or machine vision and subdivided into six principal areas: (1) sensing, (2) preprocessing, (3) segmentation, (4) description, (5) recognition, and (6) interpretation. The use of machine vision is motivated by the continuing need to increase the flexibility and scope of applications of robotic systems [*6*]. Generally, there are three tasks involved in robot vision, namely, *image transformation* or *image processing*, *image analysis*, and *image interpretation*. Image analysis is the collection of processes in which a captured image that is prepared by image processing is analyzed in order to extract information about the image and to identify objects or facts [20]. Niku further defines *image processing* as the collection of routines and techniques that improve, simplify, enhance, or, otherwise, alter an image.

9.6.1 Image Transformation/Image Processing

Image transformation involves the conversion of light energy in digital data for processing. Images are converted using equipment such as cameras, photodiodes array, charge-injection device (CID) array, and charged-coupled device (CCD) array. A 3-D image processing involves operations that require motion detection, depth measurement, remote sensing, relative positioning, and navigation. *Spatial digitization* is a process in which intensities of light at each pixel location are read while s*ampling* is a technique whereby the more pixels that are present and individually read, the better the resolution of the camera and the image. These two processes are important in image recognition and edge detection.

9.6.1.1 Histogram

A histogram is a representation of the total number of pixels of an image at each gray level. Histograms are applied in a variety of processes which can be used to determine the cutoff point when an image should be transformed to binary levels such as thresholding. These are also used for noise reduction by determining what the noisy gray level is in order to attempt to remove or neutralize the noise.

9.6.1.2 Thresholding

Thresholding is the process of dividing an image into different portions by selecting a certain grayness level as a threshold, comparing each pixel value with the threshold, and then assigning the pixel to the different portions or levels, depending on whether the pixel's grayness level is below threshold(off) or above the threshold(on) [20].

9.6.1.3 Connectivity

Connectivity is used to determine whether adjacent pixels are related to one another. It does so by ascertaining the properties of each pixel. These properties include but are not limited to pixel region, texture, and object of interest. To establish connectivity of neighboring pixels, a connectivity path must be determined (Fig. 9.13).

The 3-D connectivity between voxels (volume cells) can range from 6 to 26 while the three basic connectivity paths for 2-D image processing and analysis are *±4-connectivity or ×4-connectivity*, *H6 or V6 connectivity*, and *8-connectivity*. Refer to Fig. 9.13 for the following definitions:

a	*b*	*c*
d	*p*	*e*
f	*g*	*h*

Fig. 9.13 Neighborhood connectivity

+4-connectivity: When a pixel is analyzed only with respect to the four pixels immediately above, below, to the left, and to the right of *p* (*b*, *g*, *d*, *e*), it is a *+4-connectivity*. For a pixel $p(x,y)$, the relevant pixels are

$$(x + 1,y),(x - 1,y),(x,y - 1),(x,y - 1); \qquad (9.48)$$

×4-connectivity: A pixel's *p*'s relationship is analyzed only with respect to the four pixels immediately across from it diagonally on four sides *p*(*a*, *c*, *f*, *h*). For a pixel $p(x,y)$, the relevant pixels are

$$(x + 1,y + 1),(x + 1,y - 1),(x - 1,y + 1),(x - 1,y - 1); \qquad (9.49)$$

H6-connectivity: A pixel's *p*'s relationship is analyzed only with respect to the six neighboring pixels on the two rows immediately above and below *p* (*a*, *b*, *c*, *f*, *g*, *h*). For a pixel $p(x,y)$, the relevant pixels are

$$(x - 1,y + 1),(x,y - 1),(x + 1,y + 1),(x - 1,y - 1),(x,y - 1),(x + 1,y - 1); \qquad (9.50)$$

V6-connectivity: A pixel's *p*'s relationship is analyzed only with respect to the six neighboring pixels on the two columns immediately to the right and to the left of *p*(*a*, *d*, *f*, *c*, *e*, *h*). For a pixel $p(x,y)$, the relevant pixels are

$$(x - 1,y + 1),(x - 1,y),(x - 1,y - 1),(x + 1,y + 1),(x + 1,y),(x + 1,y - 1); \qquad (9.51)$$

8-connectivity: A pixel's p's relationship is analyzed with respect to all eight pixels surrounding it *(a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*). For a pixel $p(x,y)$, the relevant pixels are

$$(x - 1,y - 1),(x,y - 1),(x + 1,y - 1),(x - 1,y),(x + 1,y),(x - 1,y + 1),(x,y + 1),(x + 1,y + 1); \qquad (9.52)$$

9.6.1.4 Noise Reduction

In any vision system, noise (unwanted visual distortions) is always a major obstacle to overcome. Many noise reduction algorithms have been written but no matter how effective and efficient they are, 100% efficiency is almost unattainable. Noise in images is generated by inaccurate data processing from electronic components, lens scratches, dust deposits, faulty memory storage devices, illumination, and a host of others. Noise reduction and filtering techniques are generally divided into two categories-*frequency-related* and *spatial-domain* [20]. *Frequency-related* techniques operate a Fourier transform of the signal, while the *spatial-domain* techniques operate on the image at the pixel level, either locally or globally. Some techniques in noise reduction are image averaging, edge detection, neighborhood averaging, etc.

9.6.1.5 Image Averaging

Image averaging employs a technique whereby a number of images of the same scene are averaged together. Consider a case where an image acquisition is referred to as$A(x,y)$. The image can be divided into two distinct groups, the first being $I(x,y)$, which represents the desired image, and$N(x,y)$, which represents the random noise. The desired image can be found by the following equation:

$$A(x,y) = I(x,y) + N(x,y) \tag{9.53}$$

$$\frac{\sum_n A(x,y)}{n} = \frac{\sum_n I(x,y) + N(x,y)}{n} = \frac{\sum_n I(x,y)}{n} + \frac{\sum_n N(x,y)}{n} = I(x,y) \tag{9.54}$$

If the term $\sum_n \frac{N(x,y)}{n} = 0$, , then Eq. 9.54 will be written as

$$\frac{\sum_n A(x,y)}{n} = \frac{\sum_n I(x,y) + N(x,y)}{n} = \frac{\sum_n I(x,y)}{n} = I(x,y) \tag{9.55}$$

9.6.1.6 Edge Detection

A step edge in an image is an image intensity contour across which the brightness of the image changes abruptly. These are frequently associated with the projection of actual object boundaries in the scene [11]. Edges provide a compact representation of the image and can further provide an expressive representation of the salient image feature as well. Edge detection is also necessary in subsequent processes such as segmentation and object recognition. Without the techniques of edge detection to improve image quality, it is extremely difficult, if not impossible, to find overlapping parts.

9.6.1.7 Neighborhood Averaging

This technique of noise reduction has a significant disadvantage because it reduces the sharpness of the image. To partially eliminate this, other averaging filters such as the Gaussian averaging filter, also called the mild isotropic low-pass filter, can be employed. This filter improve the image quality but to a limited extent.

9.6.1.8 Hough Transform

The Hough transform is a technique used to determine the geometric relationship between different pixels on a line, including the slope of a line (Niku et al.). Consider a problem of detecting straight lines in an image from a collection of edge l (edge elements) measurements (x_i, y_i). If the edge is represented as $y = mx + b$, the measurement (x_i, y_i) provides support for the set of lines that pass through (x_i, y_i), or equivalently, for the set of values (m,b) that satisfy $y_i = mx_i + b$. The Hough transform provides a voting scheme to combine individual (x_i, y_i) measurements to obtain (m,b) [11]. The Hough-transformation expressed as an integral transformation is [25]

$$H(\mathbf{p}) = \iint_D f(x,y)\delta(g(x,y,\mathbf{p}))\mathrm{d}x\mathrm{d}y \tag{9.56}$$

This maps all the information of an edge-filtered image $\mathbf{F}(x,y)$ to the parameter space of the search pattern. The elements of $\mathbf{F}(x,y)$ are the gradient magnitude $f(x,y)$ and the gradient direction $\varphi(x,y)$. The adaptation is performed through the description $g(x,y,\mathbf{p})$, where $\mathbf{p}$ is a parameter vector that is to be determined by the feature extraction process and $H(\mathbf{p})$ is the Hough accumulator which points to corresponding features in the image.

9.6.2 Vision and Image Processing for Autonomous Robots

Vision systems for autonomous mobile robots must unify the requirements and demands of the very challenging disciplines, namely [24]

1. computer vision and image processing and
2. robotics and embedded systems.

9.6.2.1 Timeliness Constraint

Since autonomous robots in a relatively complex environment perform “simple” tasks similar to obstacle detection and object tracking, they must have image-processing software capable of calculating and processing the data at a high frequency such as 30 Hz or more. Whenever possible, image-processing operations should be executed in parallel in order to fully exploit the available resources such as dual-CPU boards and hyperthreading and multicore processor technologies.

9.6.2.2 Fixed Frame Rate Image Streams

In most cases, images are acquired at a fixed frame rate. If there is a fluctuation in data acquire in a dynamic environment, trigger devices should be built to evaluate and compensate for these changes.

9.6.2.3 Development Model

Developing a model for autonomous robots poses its own challenges. Since the robot vision is performed on live image streams, which are recorded by a moving robot platform, the development starts on sets of test images and recorded test image streams. If the application domain implies nondeterminism, or if the robot's actions affect the quality of sensor data by inducing effects like motion blur, the vision system needs to be tested rigorously in a realistic environment.

9.6.2.4 Depth Measurement with Vision Systems

There are two methods of depth measurement in a vision system.

1. The first method uses the range finders in unison with a vision system and image-processing technique. In this combination, the scenes acquired are analyzed with respect to the data received by the range finders about the distances of different portions of an environment or the location of particular objects in that environment.
2. The second method makes use of binocular or stereo vision. In this method, there are two cameras simultaneously capturing a scene which, when processing analyzed, provides a perspective view of the environment in order to ascertain the depth.

9.6.3 Position Visual Servoing (Robot Visual Control)

Visual feedback to control a robot is commonly termed *visual servoing*. In terms of manipulation, one of the main motivations for incorporating vision in the control loop was the demand for increased flexibility of robotic systems [16]. In *open-loop* robot control, initialization represents the extraction of features and characteristics that are used to directly generate the control sequence. In this case, there is no online interaction between the robot and the environment. Conversely, in the close-loop, control of a robot system vision is used as the integral sensory component consisting of tracking and control. The visual servoing system consists of *initialization*,

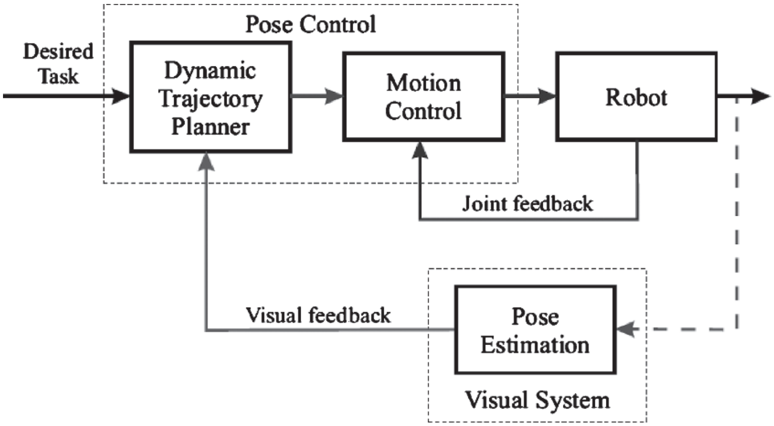


Fig. 9.14 Block scheme of a position-based visual servoing algorithm [19]

tracking, and *control*. *Initialization* occurs when the visual servoing sequence is initialized. *Tracking*, as the name implies, continuously updates the location of features and characteristics used for robot control while *robot controls* are based on the sensory input and the controls sequence of generated signals. Figure 9.14 depicts a typical position-based servoing scheme for an industrial robot.

This algorithm requires the estimation of the *pose* (position) of the target object with respect to a reference frame by using the vision system. The estimated pose is then fed back to a pose controller which performs two main operations, namely, *pose control* and *pose estimation* [19]. The main obstacle of the position-based algorithms is the real-time estimation of the pose of target objects from visual measurements.

9.6.3.1 Pose Control

Pose control is performed through an inner–outer control loop. The inner loop implements motion control, which is an independent joint control or any kind of joint space or task space control. The *dynamic trajectory planner* block (outer loop) calculates the trajectory for the end effector on the basis of the current object pose and task desired.

9.6.3.2 Pose Estimation

Pose estimation algorithm provides the measurement of the target object pose. The use of a multi-camera system requires the adoption of intelligent and computationally efficient strategies for the management of highly redundant information whereby a large number of objects image features from multiple points of view *estimation* [19].

Real-time constraints must be placed on these tasks and therefore all available visual information cannot be extracted and interpreted.

9.7 Conclusion

As discussed, there are numerous applications of autonomous robots primarily in the manufacturing and production industries. Other industries, such as the military, healthcare, and space, are gradually noticing the economic and innovative potential of applications and development of autonomous robots. Its utilization considerably improves the versatility and perceptual ability in tasks, thus expanding its relevance.

This chapter briefly highlighted the overview of autonomous robots and also illustrated some fundamental theories in robotics. Future research is geared toward improving robot autonomy by increasing its decision-making ability through the development of various innovative techniques in vision, control and sensing, and computing. Exceptional advances in robotics have been witnessed in the area of *advanced medical robotics*. Operations research methodologies are also implemented harmoniously with autonomous robots to improve efficiency and productivity in *flexible manufacturing systems*.

References

1 Agrawal, A. (1986), Robot Eye-In-Hand Using Fibre Optics, International Trends in Manufacturing Technology: Robot Sensors, Vol. 1-Vision, pp. 115–126.
2 Ansorge, D., Koller, A. (1996), Intelligent Decentralized Planning and Complex Strategies for Negotiation in Flexible Manufacturing Environments, Proceedings of the Sixth International Symposium on Robotics and Manufacturing, Vol. 6, pp. 1–6, May 28–30.
3 Asada, M., Hosoda, K., Suzuki, S. (1998), Vision-Based Behavior and Learning and Development for Emergence of Robot Intelligence, Robotics Research: The Eighth International Symposium, pp. 327–338.
4 Baird, H. S., Morris, J. (1986), Precise Robotic Assembly using Vision in the Hand, International Trends in Manufacturing Technology: Robot Sensors, Vol. 1-Vision, pp. 85–111.
5 Birk, A. (1996), Robotic Control Via Stimulus Response Learning, Proceedings of the Sixth International Symposium on Robotics and Manufacturing, Vol. 6, pp. 13–18, May 28–30.
6 Chang, T. C., Wysk, R. A., Wang, H. P. (2006), Computer-Aided Manufacturing, Pearson Education, Inc.
7 Chatila, R., Khatib, M. (1998), Interleaving Motion Planning and Execution for Mobile Robots, Robotics Research: The Eighth International Symposium, pp. 124–135.
8 Choomuang, R., Afzulpurkar, N. (2005), Hybrid Kalman Filter/Fuzzy Logic Based Position Control of Autonomous Mobile Robot, International Journal of Advanced Robotics Systems Vol. 2, No 3. pp. 197–208.
9 Corke, P. I. (1996), Visual Control of Robots: High-Performance Visual Servoing, John Wiley & Sons, New York.
10 Dillman, R., Weckesser, P. (1998), Exploration of Unknown Environments with a Mobile Robot using Multisensorfusion, Robotics Research: The Eighth International Symposium, pp. 215–236.

11 Dudek, G., Jenkin, M. (2000), Computational Principles of Mobile Robotics, Cambridge University Press.
12 Faraz, A., Payandeh, S. (2000), Engineering Approaches to Mechanical and Robotic Design for Minimally Invasive Surgeries, Kluwer Academic Publishers.
13 Fu, K. S., Gonzalez, R. C., Lee, C. S. G. (1987), Robotics: Control, Sensing, Vision and Intelligence, McGraw-Hill Book Co., New York.
14 Ikuta, K. (1998), Robot Assisted Surgery and Training for Future Minimally Invasive Therapy, Robotics Research: The Eighth International Symposium, pp. 299–316.
15 Jazar, R. N. (2007), Theory of Applied Robotics: Kinematics, Dynamics, and Control, Springer LLC, 2007.
16 Kragic, D., Christensen, H. I., Center for Autonomous Systems, Numerical Analysis and Computer, Stockholm, Sweden.
17 Lahouar, S., Zeghloul, S., Romdhane, L. (2006), Real-Time Path Planning for Multi-DoF Manipulators in Dynamic Environment, International Journal of Advanced Robotic Systems, Vol. 3, No. 2, pp. 125–132.
18 Lambert, A., Lévêque, O. (1996), Planning Autonomous Vehicle Displacement Mission in Closed Loop Form, Proceedings of the Sixth International Symposium on Robotics and Manufacturing, Vol. 6, pp. 385–392, May 28–30.
19 Lippiello, V., Siciliano, B. (2006), Position-Based Visual Servoing in Industrial Multi-Robot Cells Using a Hybrid Camera Configuration, IEEE, Revised August.
20 Niku, S. B. (2001), Introduction to Robotics Analysis, Systems, Applications, Prentice-Hall International, Upper Saddle River, NJ.
21 Pinson, L.J. (1986), Robot Sensors: An Evaluation of Imaging Sensors, International Trends in Manufacturing Technology: Robot Sensors, Vol. 1-Vision, pp. 15–63.
22 Rizzi, A. A., Hollis, R. L. (1998), Opportunities for Increased Intelligence and Autonomy in Robotic Systems for Manufacturing, Robotics Research: The Eighth International Symposium, pp. 141–151.
23 Topalova, I. (2005), Modular Adaptive System Based on a Multi-Stage Neural Structure for Recognition of 2D objects of Discontinuous Production, International Journal of Advanced Robotics Systems, Vol. 2, No 1. pp. 45–51.
24 Utz, H., Kaufmann, U., Mayer, G., Kraetzschmar, G.K. (2006), VIP-A Framework-Based Approach to Robot Vision, International Journal of Advanced Robotics Systems, Vol. 3, No 1. pp. 67–72.
25 Vincze, M., Hager, G. D. (2000), Robust Vision for Vision-Based Control of Motion, The Institute of Electrical and Electronics Engineers, Inc., New York.

Chapter 10
Modular Design

Sa'Ed M. Salhieh [1] **and Ali K. Kamrani** [2]

Abstract Modular design aims to subdivide a complex product into smaller parts (modules) that are easily used interchangeably. Examples of modularly designed items are vehicles, computers, and high-rise buildings. Modular design is an attempt at getting both the gains of standardization (high volume normally equals low manufacturing costs) and the gains of customization. The concept of modularity can provide the necessary foundation for organizations to design products that can respond rapidly to market needs and allow the changes in product design to happen in a cost-effective manner. Modularity can be applied to the design processes to build modular products and modular manufacturing processes.

10.1 Modularity

Modularity aims to identify the independent, standardized, or interchangeable units to satisfy a variety of functions. Modularity can be applied in the areas of product design, design problems, production systems, or all three. It is preferable to use modular design in all three types at the same time; this can be done by using a modular design process to design modular products and to produce them using a modular production system or modular manufacturing processes.

10.1.1 Modularity in Products

Modular products are products that fulfill various overall functions through the combination of distinct building blocks or modules, in the sense that the overall function performed by the product can be divided into subfunctions implemented

---

[1] Assistant Professor, Department of Industrial Engineering, University of Jordan, Amman, 11942, Jordan

[2] Associate Professor, Department of Industrial Engineering, University of Houston Houston, Texas 77004, USA

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5, © Springer Science + Business Media, LLC 2008

by different modules or components [13]. Product modularity has been analyzed as a form of product architecture that allows a one-to-one correspondence between physical structures and functional structures, as opposed to integral architectures where the functional elements map to a single or very small number of physical elements [15]. Hand tools are considered a good example of integral products, where several functions are mapped to a single physical structure, that is, the tool itself. A personal computer exemplifies a modular product in which a wide range of functions are fulfilled by utilizing a wide range of interchangeable physical structures such as hard drives, CD-ROMs, and motherboards.

The term modularity in products is used to describe the use of common units to create product variants. That is, modularity in products is based on the idea that a complex product could be decomposed into a set of independent components. This decomposition allows the standardization of components and the creation of product variants. Components used to create modular products have functional, spatial, and other interface characteristics that fall within the range of variations allowed by the specified standardized interfaces of a modular product. Mixing and matching different modular components creates a large number of modular products, where each product would have a distinct combination of the modular components, resulting in the creation of products with distinctive functionalities, features, and performance levels.

10.1.2 Modularity in Design Problems

Most design problems can be broken down into a set of easy-to-manage simpler subproblems. Sometimes complex problems are reduced into easier subproblems, where a small change in the solution of one subproblem can lead to a change in other subproblems' solutions. This means that the decomposition has resulted in functionally dependent subproblems. Modularity focuses on decomposing the overall problem into functionally independent subproblems, in which interaction or interdependence between subproblems is minimized. Thus, a change in the solution of one problem may lead to a minor modification in other problems, or it may have no effect on other subproblems.

10.1.3 Modularity in Production Systems

Modularity in production systems aims at building production systems from standardized modular machines. The fact that a wide diversity of production requirements exists has led to the introduction of a variety of production machinery and a lack of agreement on what the building blocks should be. This means that there are no standards for modular machinery. In order to build a modular production system, production machinery must be classified into functional groups from which a selection of a modular production system can be made to respond to different production requirements. Rogers [11] classifies production machinery into four basic groups of “primitive” production elements. These are process machine primitives, motion units, modular fixtures, and configurable control units. It is argued that if a selection is made from these four categories, it will be possible to build a diverse range of efficient, automated, and integrated production systems.

10.2 Modular Systems Characteristics

10.2.1 Modules Types

Modular systems are built from independent units or modules. Two major categories of modules are identified, namely, *function modules* and *production modules* [12]. Function modules are designed to accomplish technical functions independently or in combination with other modules. Production modules are designed on the basis of production considerations alone and are independent of their function. Function modules can be classified on the basis of various types of functions reoccurring in a modular system that can be combined as subfunctions to implement the different overall function (Fig. 10.1). These functions are basic, auxiliary, special, adaptive, and customer specific [9].

- *Basic Functions.* These are functions that can fulfill the overall function simply or in combination with other functions. Basic functions are not variable in principle and they are implemented in *basic modules*.

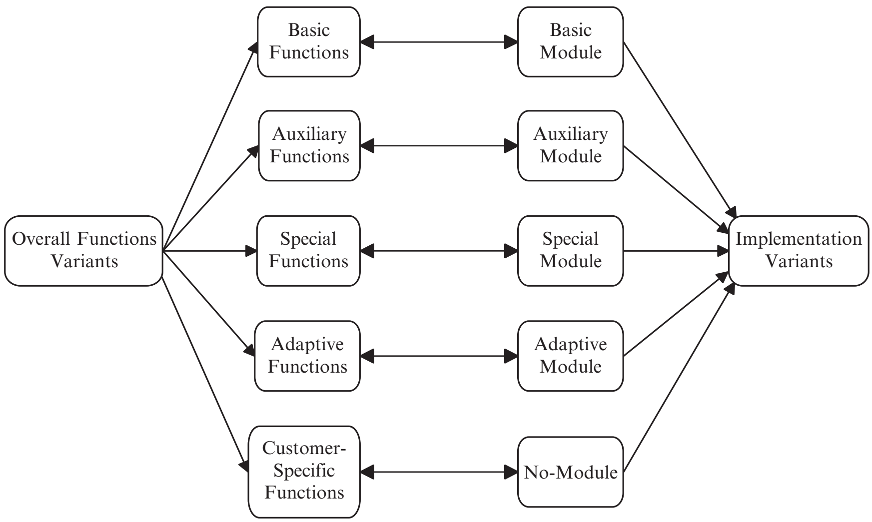


Fig. 10.1 Function and module types

- *Auxiliary Functions.* These are implemented using auxiliary modules in accordance with basic modules.
- *Special Functions.* These are task-specific subfunctions that may not appear in all overall function variants and are implemented by *special modules*.
- *Adaptive Functions.* These are the functions that permit the adaptation of a part or a system to other products or systems. They are implemented by *adaptive modules* that allow for unpredictable circumstances.
- *Customer-Specific Functions.* These are functions that are not provided by the modular system, and they are implemented by *non-modules* which must be designed individually. If they are used, the result is a mixed system that combines modules and non-modules.

10.2.2 Modularity Types

Product modularity depends on the similarity between the physical and the functional architecture of a design, and on the minimization of the incidental interactions between the physical components that comprise the modules. The nature of the interactions between the modules has been used to categorize product modularity into two major categories of modularity [6, 15, 16]:

- Function-based modularity is used to partition the functionalities of a product and describe how these functions are distributed.
- Manufacturing-based modularity relates to the manufacturing processes and the assembly operations associated with a product.

10.2.2.1 Function-Based Modularity

Four classifications of function-based modularity are defined as follows.

1. *Component-Swapping Modularity:* Different product variants belonging to the same product family are created by combining two or more alternative types of components with the same basic component or product. Figure 10.2 illustrates the swapping modularity in which two alternative components (the small rectangular block and the triangular) are combined with the same basic component (the big block), forming product variants belonging to the same product family.

An example of component-swapping modularity in the computer industry is illustrated by matching different types of CD-ROMs, monitors, and keyboards with the same motherboard. This allows for different models of computers to be implemented.

2. *Component-Sharing Modularity*: In this category, different product variants belonging to different product families are created by combining different modules sharing the same basic component. Component-sharing is considered

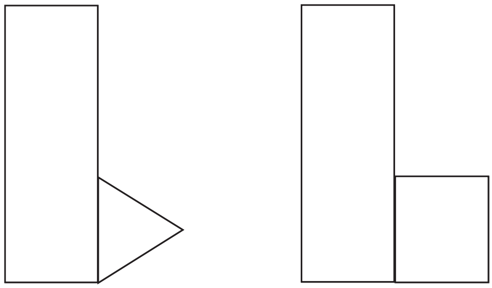


Fig. 10.2 Component-swapping modularity

the complementary case to component-swapping. Component-sharing and component-swapping modularity are identical except that swapping involves the same basic product using different components and sharing involves different basic products using the same component. The difference between them lies in how the basic product and components are defined in a particular situation. Figure 10.3 shows two different basic components (the block and the triangular) sharing the same component (the circle). Component-sharing modularity in the computer industry is represented by the use of the same power cord, monitor, or microprocessor in different product (computer) families.

3. *Fabricate-to-Fit Modularity:* One or more standard components are used with one or more infinitely variable additional components. Variation is usually associated with physical dimensions that can be modified. Figure 10.4 illustrates a component with variable length (the block) that can be combined with two standard components (the triangular) forming product variants. A common example of this kind of modularity is cable assemblies in which two standard connectors can be used with an arbitrary length of cable.
4. *Bus Modularity:* This type of modularity occurs when a module can be matched with any number of basic components. Bus modularity allows the number and location of basic components in a product to vary. Bus modularity is illustrated in Fig. 10.5. An example of bus modularity is a computer where different input and output units, in addition to different types of mice, RAMs, and hard drives, can exist and vary in both their location and their number.

10.2.2.2 Manufacturing-Based Modularity

Four manufacturing-based modularity classes can be defined as follows.

1. OEM (Original Equipment Manufacturer) modules are group of components that are grouped together because a supplier can provide them at a less expense than if they were to be developed in-house. For example, tires in cars are OEM modules.
2. Assembly modules are groups of components that are grouped together because they solve related functions and are bundled together to ease assembly. For

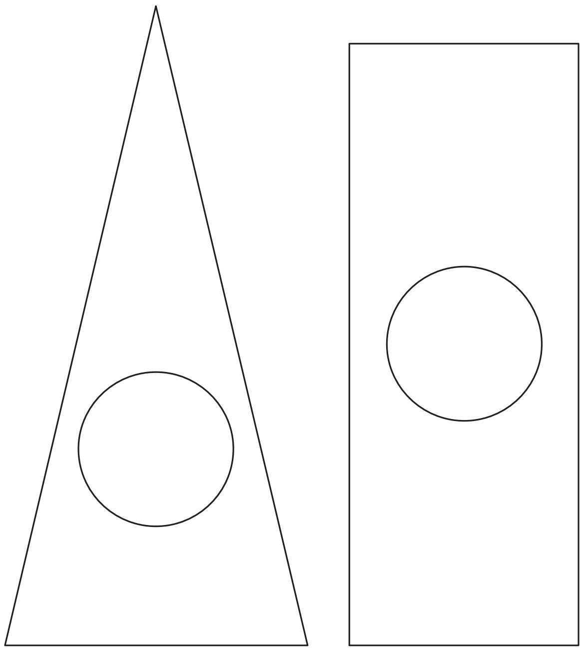


Fig. 10.3 Component-sharing modularity

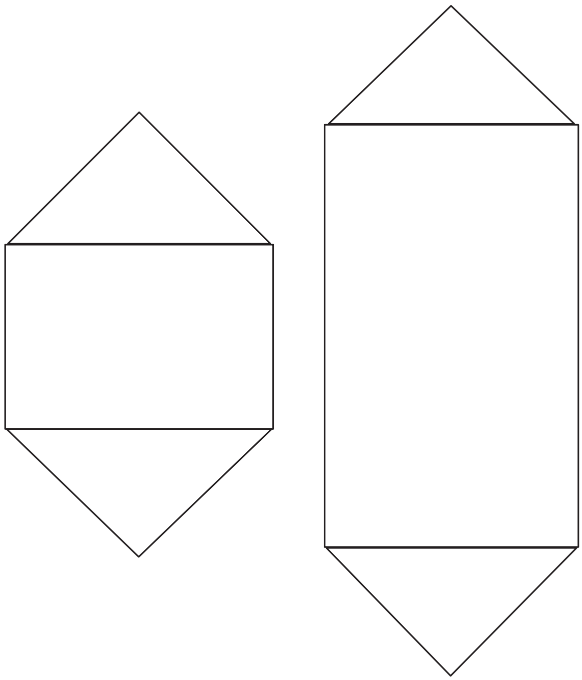


Fig. 10.4 Fabricate-to-fit modularity

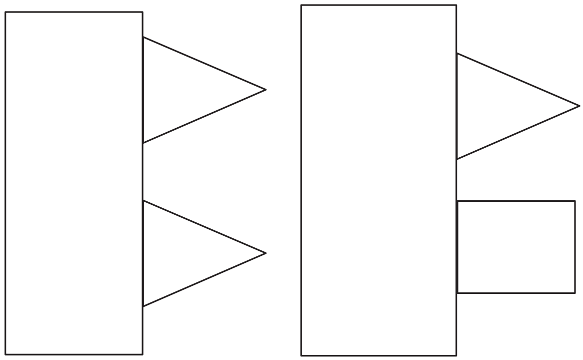


Fig. 10.5 Bus modularity

example, the dial buttons and the associated electronic circuit in a telephone are all bundled together as a subassembly when making telephones.
3. Sizable modules are components that are exactly the same except for their physical scale. Sizable modules are manufactured using the same exact operations and machine. Lawn mower blades are an example of sizeable modules.
4. Conceptual modules are modules that deliver the same functions but have different physical embodiments. Conceptual modules can lead to a significant change in the manufacturing operations without affecting the functionality of the product. For example, designers may use a gearbox to reduce the speed delivered from a motor to a pump, or they could use a chain-sprocket system to deliver the same function.

10.3 Modular Systems Development

In general, modular systems can be developed by decomposing a system into its basic functional elements, mapping these elements into basic physical components, then integrating the basic components into a modular system capable of achieving the intended functions. This approach faces two important challenges [10]: (1) Decomposition: Finding the most suitable set of subproblems may be difficult. (2) Integration: Combining the separate subsystems into an overall solution may also be difficult. To fully comprehend the underlying foundations of modular systems development, decomposition categories are further discussed.

10.3.1 Decomposition Categories

System decomposition is expected to result in two benefits [10]: (1) Simplification: Decomposing large systems into smaller ones will lead to a reduction in the size of the problem that needs to be solved, which will make it easier to manage. (2) Speed: Solving smaller problems concurrently (parallel solutions) will reduce the time needed to solve the overall problem. Decomposition methods can be categorized according to the area into which they are being applied, namely, product decomposition, problem decomposition, and process decomposition [*8*].

10.3.1.1 Product Decomposition

Product decomposition can be performed at various stages of the design process and can be defined as the process of breaking the product down into physical elements from which a complete description of the product can be obtained. Two approaches are used in product decomposition, *product modularity* and *structural decomposition*.

1. *Product Modularity*

   Product modularity is the identification of independent physical components that can be designed concurrently or replaced by predesigned components that have similar functional and physical characteristics. Product modularity relies on the lack of dependency between the physical components. The computer industry provides an excellent example of modular products, where the major components of the computer are manufactured by many different suppliers allowing the manufacturers of microprocessors to choose from a wide library of products.

2. *Structural Decomposition*

   The system is decomposed into subsystems, and those are further decomposed into components leading to products, assemblies, subassemblies, and parts at the detailed design stage. The decomposition is represented in a hierarchy structure that captures the dependencies between subsystems.

10.3.1.2 Problem Decomposition

For centuries, complex design problems were handled by breaking them into simpler, easy-to-handle subproblems. Problem decomposition should continue until basic independent products or units are reached. The interaction between the basic products should be identified and introduced as constraints imposed by higher subproblems. Problem decomposition is divided into *requirements decomposition*, *constraint–parameter decomposition*, and *decomposition-based design optimization*.

1. *Requirements Decomposition*

   Requirements represent an abstraction of the design problem, starting with the overall requirement (general demand) and ending with the specific requirements (specific demands). The ability to meet a requirement is given by a design function. The requirements decompositions and their relationships to the corresponding functions are represented in a tree diagram (Fig. 10.6), where specific requirements are mapped into specific functions.

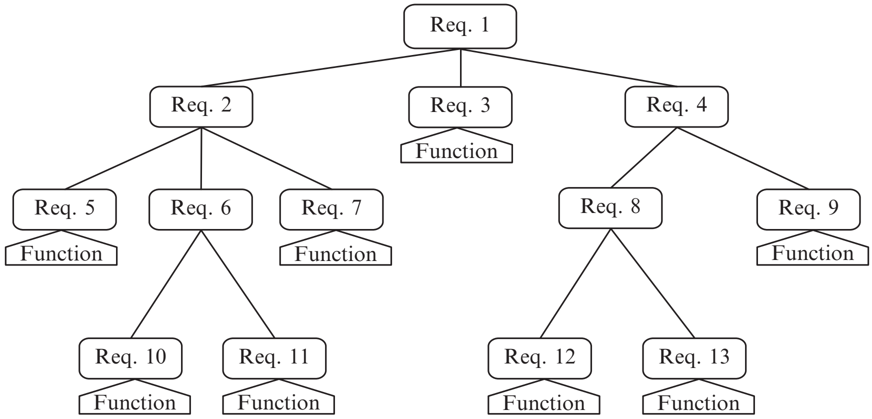


Fig. 10.6 Requirements decomposition

2. *Constraint–Parameter Decomposition*

   The parameters describe the features (quantitative or qualitative data) of the product, while the constraints define the ranges of values assigned to parameters that are defined by product requirements. The problem structure is represented in an incidence matrix [8]. The incidence matrix is decomposed by grouping all nonempty elements in blocks at the diagonal. It is preferable that the blocks be mutually separable (independent). In some cases, overlapping between variables or constraints may occur.

   The design of a ball bearing is used to illustrate the decomposition [8]. The parameters are listed in Table 10.1 and the constraints are shown in Table 10.2. The constraint–parameter incidence matrix is shown in Fig. 10.7. The decomposed matrix is shown in Fig. 10.8.

3. *Decomposition-Based Design Optimization*

   The decomposition of a large complex design problem into smaller independent subproblems facilitates the use of mathematical programming techniques to solve and optimize the subproblems [3, 4]. The solutions are integrated to provide an overall solution. The objective is to decompose a complex system into multilevel subsystems in a hierarchical form (Fig. 10.9), in which a higher-level subsystem controls or coordinates the subsystems at the lower level. The subsystems are solved independently at the lower level. The objective at the higher level is to coordinate the action of the first level to ensure that the overall solution is obtained.

10.3.1.3 Process Decomposition

Process decomposition is the decomposition of the entire design process, starting with the need recognition and ending with the detail design. The activities in the design process are modeled in a generic manner independent of the specific product being

Table 10.1 Ball bearing design parameters

Parameter	Description	Parameter	Description
d_e	Pitch diameter	β_I	Free contact angle
d_o	Outer-race diameter	r_o	Outer-race curvature
d_i	Inner-race diameter	r_i	Inner-race curvature
P_d	Diametral clearance	P_e	Free endplay
d	Rolling-element diameter	s	Shoulder height
I	Race conformity ratio	θ	Shoulder angle height
r	Race curvature radius	R	Curvature sum
B	Total conformity	R_x	x direction effective radius
I_o	Outer-race conformity	R_y	y direction effective radius
I_i	Inner-race conformity	Γ	Curvature difference
D	Race curvature distance	β	Contact angle

Table 10.2 Ball bearing design constraints

C_1	$d_e = \frac{1}{2}(d_o + d_i)$	C_7	$P_e = 2DSin\ \beta_f$
C_2	$P_d = d_o - d_i - 2d$	C_8	$S = r(1 - \cos\theta)$
C_3	$f = \frac{r}{d}$	C_9	$\frac{1}{R} = \frac{1}{R_x} + \frac{1}{R_y}$
C_4	$B = f_o + f_i - 1$	C_{10}	$\Gamma = R = \left(\frac{1}{R_x} - \frac{1}{R_y}\right)$
C_5	$D = Bd$	C_{11}	$R_x = d(d_e - d\cos\beta)/2d_e$
C_6	$\beta_f = \arccos\frac{r_o + r_i - \frac{1}{2}(d_o - d_i)}{r_o + r_i - d}$	C_{12}	$R_y = \frac{f_i d}{(2f_i - 1)}$

	d_e	d_o	d_i	P_d	d	I	r	B	I_o	I_i	D	β_I	r_o	r_i	P_e	s	θ	R	R_x	R_y	Γ	β
C_1	*	*	*																			
C_2		*	*	*	*																	
C_3					*	*	*															
C_4								*	*	*												
C_5					*			*			*											
C_6		*	*		*							*	*	*								
C_7											*	*			*							
C_8							*									*	*					
C_9																		*	*	*		
C_{10}																		*	*	*	*	
C_{11}	*				*														*			*
C_{12}					*					*										*		

Fig. 10.7 Ball bearing design constraint–parameter incidence matrix

	l	r	s	θ	d_e	d_o	d_i	β	P_d	r_o	r_i	l_o	l_i	R_y	R	Γ	D	P_e	d	β_l	R_x	B
C_3	*	*																	*			
C_8		*	*	*																		
C_1					*	*	*															
C_{11}					*			*											*		*	
C_2						*	*		*										*			
C_6						*	*			*	*								*	*		
C_4												*	*									*
C_{12}													*	*					*			
C_9														*	*						*	
C_{10}														*	*	*					*	
C_5																	*		*			*
C_7																	*	*		*		

Fig. 10.8 Decomposed constraint–parameter incidence matrix

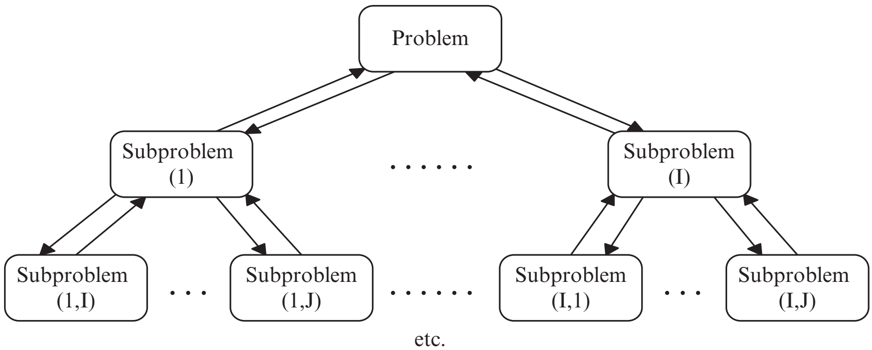


Fig. 10.9 Hierarchical decomposition of a complex system

designed. Three perspectives of process decomposition were recognized. These are *product flow perspective*, *information flow perspective*, and *resource perspective*.

1. *Product Flow Perspective*

   Design activities required to translate customer requirements into a detailed design of products are the focus of this perspective. The design activities are modeled as blocks with identified inputs and outputs (the output of one activity becomes the input of another activity). The decomposition tries to eliminate redundant activities and reorganize other activities to be performed concurrently, which will eventually reduce the product development time.

2. *Information Flow Perspective*

   Analysis of the precedence constraints between the design activities is the main concern of this perspective. Precedence constraints are utilized to generate the required information needed to build supporting databases and communication networks and to schedule design activities, all concurrently.

3. *Resource Perspective*

The resources provide activities with a mechanism for transforming inputs to outputs. In this perspective two types of constraints are considered:

- *External resource constraints*, in which the resource used by the activity is generated by an activity or resource that is external to the design process.
- *Internal resource precedence constraints*, in which the resource is developed in the design process and used by other activities.

10.3.2 Component Grouping into Modules

After decomposing the system into its basic components or elements, a modular system should be constructed by integrating the basic similar elements based on a criteria set by the product design team. A modular system can be thought of as an integration of several functional elements that, when combined, perform a different function than their individual one. The similarity between the physical and functional architecture of the design must be used as a criteria for developing modular systems. Another criterion that must be used is the minimization of the degree of interaction between physical components. The degree of interaction between physical elements is an important aspect of modularity, which must be identified, minimized, or eliminated. The strength of a modular system design can be measured by the weakness of the interactions or the interfaces between its components.

Grouping objects (i.e., components, parts, or systems) into groups based on the object features has been done using Group Technology (GT) approaches. GT is defined as the realization that many problems are similar, and that by grouping similar problems, a single solution can be found to a set of problems, thus saving time and effort [1, 5, 7]. Similar components can be grouped into design families and new designs can be created by modifying an existing component design from the same family. Objects grouping or cluster analysis in GT is concerned with grouping parts into part families and machines into machine cells [2]. A number of algorithms and methods are available for clustering parts and machines such as the following [14]:

- The rank order clustering algorithm
- The modified rank order clustering algorithm
- The bond energy algorithm
- The cluster identification algorithm
- The extended cluster identification algorithm
- Similarity coefficient-based clustering
- Mathematical programming-based clustering

10.4 Modular Product Design

Modular product design is an important form of strategic flexibility, that is, flexible product designs that allow a company to respond to changing markets and technologies by rapidly and inexpensively creating product variants derived from different combinations of existing or new modular components. Kamrani and Salhieh (2002) [6] developed a four-step methodology for the development of modular products as follows.

10.4.1 Needs Analysis

This step includes gathering the market information required to identify customer needs, and arranging the identified needs into groups and finally prioritizing the needs according to their importance.

10.4.2 Product Requirements Analysis

Product requirements are identified on the basis of the results of needs analysis. The product requirements are classified into three classes as follows.

- *Functional objectives* needed to meet the customer's primary needs.
- *Operational functional* requirements that impose both functional and physical constraints on the design.
- *General functional requirements* (*GFRs*) that satisfy customers' secondary needs, which could form a critical factor for the customer when comparing different competitive products that accomplish the same function. GFRs should be weighted with respect to their importance.

10.4.3 Product/Concept Analysis

Product/concept analysis is the decomposition of the product into its basic functional and physical elements. These elements must be capable of achieving the product's functions. Functional elements are defined as the individual operations and transformations that contribute to the overall performance of the product. Physical elements are defined as the parts, components, and subassemblies that ultimately implement the product's function. Product concept analysis consists of product physical decomposition and product functional decomposition. In product physical decomposition, the product is decomposed into its basic physical components which, when assembled together, will accomplish the product function. Physical decomposition should result in the identification of basic components that must be designed or selected to perform the product function. Product functional decomposition describes the product's overall functions and identifies components' functions. Also, the interfaces between functional components are identified.

- *Product Physical Decomposition.* The product is decomposed into subsystems and/or subassemblies capable of achieving the product function. The decomposition process should continue until basic physical components are reached.
- *Product Functional Decomposition.* Functional decomposition should aim at representing the intended behavior (the functions) of a product and its parts. A function could be implemented by a single physical element (component) or by a combination of elements arranged in a specific manner. Functional components are arranged according to several logical considerations that will ensure the accomplishment of their intended combined function. The logical arrangement is called a working principle which defines the mode of action that the product/system will perform on the inputs to reach the output state. To analyze the product function, the overall function of the product should be conceptualized into an action statement (verb–noun form). Then, the overall function is broken into subfunctions, and those are further decomposed into lower-level functions. This functional breakdown is continued until a set of functions that could be achieved by available components is reached. At this point, functions are mapped into components, and components are arranged forming subassemblies leading to an overall assembly that will ultimately accomplish the overall function.

10.4.4 Product Concept Integration

Basic components resulting from the decomposition process are arranged in modules and integrated into a functional system. The manner by which components are arranged in modules will affect the product design. The resulting modules can be used to structure the development teams needed. Following are the steps associated with product integration.

10.4.4.1 Identify System-Level Specifications

System-level specifications (SLS) are the one-to-one relationship between components with respect to their functional and physical characteristics. Functional characteristics are a result of the operations and transformations that components perform in order to contribute to the overall performance of the product. Physical characteristics are a result of the components' arrangements, assemblies, and geometry that implement the product function. A general guideline for identifying the relationships can be presented as follows:

1. *Functional characteristics*
   (a) Identify the main function(s), based on the functional decomposition.
   (b) Identify the required operations and transformations that must be performed in order to achieve the function based on the function flow diagram.
   (c) Document the operations and transformations.
   (d) Categorize operations and transformations into a hierarchy structure.

2. *Physical characteristics*

   (a) Identify any physical constraints imposed on the product based on the requirement analysis.
   (b) Identify possible arrangements and/or assemblies of the components, based on previous experiences, previous designs, engineering knowledge, or innovative designs/concepts.
   (c) Document possible arrangements and/or assemblies.
   (d) Categorize arrangements and assemblies into a hierarchy structure.

Physical and functional characteristics, forming the SLS, are arranged into a hierarchy of descriptions that begins by the component at the top level and ends with the detailed descriptions at the bottom level. Bottom-level descriptions (detailed descriptions) are used to determine the relationships between components, 1 if the relationship exists and 0 otherwise. This binary relationship between components is arranged in a vector form, "System-Level Specifications Vector" (SLSV). Figure 10.10 illustrates the hierarchical structure of the physical and functional characteristics.

10.4.4.2 Identify the Impact of the System-Level Specifications on the General Functional Requirements

SLS identified in the previous step affect the GFRs in the sense that some specifications may help satisfy some GFRs, while other specifications might prevent the implementation of some desired GFRs. The impact of the SLS on GFRs should be clearly identified. This will help in developing products that will meet, up to a satisfactory degree, the GFRs stated earlier. The impact will be determined on the basis of the following:

-1	:	Negative impact
0	:	None
+1	:	Positive impact

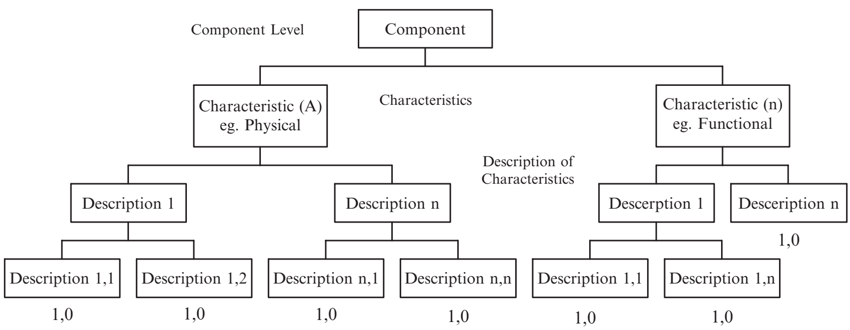


Fig. 10.10 System-level specification decomposition hierarchy

Table 10.3 GFR versus SLS

System-level specifications	General functional requirements		
	FR (1)	FR (2)	FR (m)
SLS (1)	–1	1	0
SLS (2)	.	.	.
SLS (n)	1	0	1

A negative impact represents an undesired effect on the GFRs such as limiting the degree to which the product will meet the general requirement, or preventing the product from implementing the general requirement. While a positive impact represents a desired effect that the SLS will have on the general requirements, such SLS will ensure that the product will satisfy the requirements and result in customer satisfaction. An SLS is said to have no impact if it neither prevents the implementation of the GFR nor helps satisfying the GFR. An example of the SLS impact on the GFRs is shown in Table 10.3.

For example, the SLS (1) have a negative impact on the FR (1), positive impact on the FR (2), and no impact on the FR (m).

10.4.4.3 Calculate Similarity Index

The degree of association between components should be measured and used in grouping components into modules. This can be done by incorporating the GFR weights, in addition to the SLSVs and their impacts on the GFRs to provide a similarity index between components. The general form of the similarity index is as follows:

$$(S)_{1\times 1} = \overset{\text{SLSV}}{(1} \quad \overset{(C_1\&C_2)}{0} \quad . \quad . \quad a_n)_{1\times n} \times \overset{\text{SLS}\,\&\,\text{FR}_s}{\begin{pmatrix} 1 & . & . & . & b_{1,m} \\ 0 & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ b_{n,1} & . & . & . & b_{n,m} \end{pmatrix}_{n\times m}} \times \overset{\text{Weights for Fl}}{\begin{pmatrix} 1 \\ 0.9 \\ . \\ . \\ c_{m\times 1} \end{pmatrix}_{m\times 1}}$$

The similarity indexes associated with components are arranged in a component versus component matrix as shown below:

	C1	C2	C3	–	Cn
C1	X	$S_{1\times 2}$	$S_{1\times 3}$	–	$S_{1\times n}$
C2		X	$S_{2\times 3}$	–	$S_{2\times n}$
C3			X	–	$S_{3\times n}$
–				X	–
Cn					X

10.4.4.4 Group Components into Modules

Components with high degree of association should be grouped together in design modules. This can be accomplished by using an optimization model that maximizes the sum of the similarities. The optimization model will identify independent modules that can be designed simultaneously. The model is Np-Hard.

Heuristic algorithms have been used as an alternative technique for solving Np-Hard problems. Modularity decomposition problem could benefit from these algorithms for finding solution in less time. This proposed method for decomposing similarity matrices in modularity is based on Network Flows and Optimization. The main reason for the application of network optimization as base structure is its ability in coupling deep intellectual contents with a remarkable range of applicability. Many combinatorial problems which are innately hard can be solved by transforming to network concepts.

This specification makes us to define our problem as a graph acceptable in Network Flows rules to have the opportunity to solve this decomposition by Network algorithms. After calculation of similarity measure, nodes and edges are defined as follows:

- *Node*: Each component represents a node in our proposed graph.
- *Edge*: Relationship between any two components represents an edge such that if component *i* has similarity index more than 0 with component *j*, there exists an edge between them.
- *Flow of each edge*: The similarity index between each two components is the flow of the edge between them (see Fig 10.11).

The objective function associated with this network is to find modules such that each module has the maximum amount of collective similarity indexes. It means that it is necessary to find modules created by group of components which are connected with each other by the largest similarity indexes. Each component can be assigned to just one module and each module must contain most similar components to each other. Based on this objective, Max Flow algorithms are used for solving this problem. In a capacitated network, it is required to send as much flow between start and end node. This concept has been applied in other engineering applications such as matrix rounding problem and feasible flow problems.

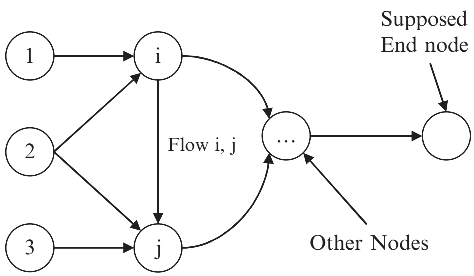


Fig. 10.11 Graph for a decomposed product

In this problem, the maximum flow for the main graph is determined. This establishes the solution for the first module that contains components (nodes) in maximum flow. Components which are assigned to the first module will be removed and a new graph will be created and this graph for maximum flow is solved to find the second module. The algorithm will continue until all possible modules are assigned. For the max flow algorithm, augmenting path algorithm or preflow-push algorithm will be used on the basis of network specifications.

The preflow-push algorithm is one of the most powerful and flexible algorithms for max flow problems. The preflow-push algorithms maintain a preflow at each intermediate stage. Active nodes in this algorithm are nodes that have positive excess. Because this algorithm attempts to achieve feasibility and in a preflow-push algorithm, the presence of active nodes indicates that the solution is infeasible, the basic operation of this algorithm is to select an active node and try to remove its excess by pushing flow to its excess.

A maximum preflow is defined as a preflow with the maximum possible flow into the sink. This algorithm is a polynomial algorithm. The sample of the pseudocodes for the solution methodology is as follows:

Network Set Begin

x = 1
n = number of components
While *each component has assigned to a module*
Design graph of components.
Use preflow-push algorithm to find the maximum flow from each node to node t.
Assign components in the maximum flow with similarity more than 1 to a module named module x.
Similarity of module = Sum (similarity of components)
x = x + 1.
End while.
Use selecting Procedure
End *Procedure.*
Selecting Procedure Begin
y = 1
While *n < > 0*
Choose the module with biggest Similarity
Eliminate components in that module from total components.
n = n – Eliminated components
y = y + 1
End while
y = number of modules
Write the selected modules as final modules.
End *Procedure.*

Matlab® software is used for the implementation of the preflow-push max flow algorithm. Using this algorithm and Matlab Code for preflow-push max flow algorithm with a four-gear speed reducer with 17 components, the solution is as follows:

- Module 1: Gear 1, Shaft 1, Bearing 1, Bearing 2, Key 1, and Gear 2.
- Module 2: Gear 2, Gear 3, Shaft 2, Bearing 3, Bearing 4, Key 2, and Key 3.
- Module 3: Gear 4, Shaft 3, Bearing 5, Bearing 6, and Key 4.

10.5 The Benefits of Product Modularity

One of the most important benefits of promoting modularity is the need to allow a large variety of products to be built from a smaller set of different modules and components. The result is that any combination of modules and components, as well as the assembly equipment, can be standardized. The benefits of product modularity also include the following:

1. *Reduction in Product Development Time*

   Modularity relies on dividing a product into components with clear definition of the interfaces. These interfaces permit the design tasks to be decoupled. This decoupling results in a reduction in the design complexity and enables design tasks to be performed concurrently, which will eventually reduce the product development time.
2. *Customization and Upgrades*

   Modular products accomplish customer requirements by integrating several functional components interacting in a specific manner. This integration allows products to be improved and upgraded by using more efficient components that can perform the required functions effectively. In addition, components can be replaced by custom-made ones to fulfill different functions.
3. *Cost Efficiencies Due to Amortization*

   Modular components are used in several product lines, which infer that their production volumes are higher. This will allow the amortization of the development expenses over a large number of products.
4. *Quality*

   Modularity allows production tasks to be performed simultaneously. Thus, independent components can be produced and tested separately before they are integrated into a modular product. This will help build quality into the product.
5. *Design Standardization*

   Modular design facilitates design standardization by identifying the component functions clearly and minimizing the incidental interactions between a component and the rest of the product.
6. *Reduction in Order Lead Time*

   Modular products can be made by combining standardized and customized components. This allows standard components to be inventoried, and then customization can be focused on the differentiating components. Also, modular products can be a combination of standard components, that is, the same standard components (usually kept in inventory) are integrated in different ways to form a variety of products that can respond to customer requirements.

References

1. Amirouche, F., Computer Aided Design and Manufacturing, Prentice Hall: Englewood Cliffs, New Jersey, 1993.
2. Erhorn, C., and Stark, J., Competing by Design: Creating Value and Market Advantage in New Product Development, Essex Junction, VT, 1994.
3. Finger, S., and J. R. Dixon, A Review of Research in Mechanical Engineering Design, Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes, *Research in Engineering Design*, Vol. 1:51–67, 1989.
4. Johnson, R. C., and R. C. Benson, A Basic Two-Stage Decomposition Strategy for Design Optimization, *Transactions of the ASME*, Vol. 106, September 1984.
5. Kamrani, A., A Methodology for Manufacturing Cell Design in a Computer Integrated Manufacturing Environment, Published Ph.D. Dissertation, University of Louisville, 1991.
6. Kamrani, A., and S. Salhieh, Product Design for Modularity, 2nd Edition, Kluwer Academic Publishers, 2002.
7. Kamrani, A. K., Modular Design Methodology for Complex Parts, Industrial Engineering Research Conference, Miami Beach, Florida, May 1997.
8. Kusiak, A., and N. Larson, Decomposition and Representation Methods in Mechanical Design, *Transactions of the ASME*, Vol. 117, June 1995.
9. Pahl, G., and W. Beitz, Engineering Design: A Systematic Approach, 3rd Edition, Springer-Verlag: New York, 2007.
10. Pimmler, T. U., and S. D. Eppinger, Integration Analysis of Product Decompositions, Design Theory and Methodology—DTM'94, DE-Vol. 68, ASME, 1994.
11. Rogers, G. G., and L. Bottaci, Modular Production Systems: A New Manufacturing Paradigm, *Journal of Intelligent Manufacturing*, Vol. 8, No. 2, pp. 147–156, April 1997.
12. Roozenburg, N. F. M., Product Design: Fundamentals and Methods, John Wiley & Sons: Chichester, New York, 1995.
13. Salhieh, S., and A. Kamrani, Macro Level Product Development Using Design for Modularity, *Robotics and Computer Integrated Manufacturing Journal*, No. 15, pp. 319–329, 1999.
14. Singh, N., and D. Rajamani, Cellular Manufacturing Systems: Design, Planning, and Control, Chapman & Hall, Norwell, Massachusetts, USA 1996.
15. Stone, R., and K. Wood, A heuristic method for identifying modules for product architectures, *Design Studies*, Vol. 21, pp. 5–31, 2002.
16. Ulrich, K., and K. Tung, Fundamentals of product modularity, In *Issues in Design/Manufacture Integration 1991*, pp. 73–79. A. Sharon Ed., ASME: New York, NY, 1991.

Chapter 11
Manufacturing Complexity Analysis: A Simulation-Based Methodology

Ali K. Kamrani[1] **and Arun Adat**[2]

Abstract Variability in products is driven by the customer and pushes the manufacturer to offer product variants by mass customization. Companies that offer product variety while maintaining competitive cost and quality will gain a competitive edge over other companies in today's market. As the automobile industry adapts to the mass customization strategy, it would require the ability to conduct early design, development, and manufacturing trade-offs among competing objectives. An analytical approach is then required to manage the complexity and the risk associated with this environment. This chapter will present a set of simulation-based methodologies for measuring complexity. The developed methodologies will assist designers in analyzing and mitigating the risks associated with product variety and its impact on manufacturing complexity.

11.1 Introduction

Agility in manufacturing is one of the many profound qualities companies must possess to succeed in a turbulent business environment. Businesses worldwide are constantly faced with the challenge of offering the ideal product variety in their supply chain because of the conflicting interests between manufacturing and marketing. Manufacturing prefers a single product in large volumes (mass production) for benefiting on the economies of scale, while marketing prefers numerous product variants (mass customization) to target a wide spectrum of the market. Although mass customization is ideally desirable from a customer service perspective, it is often accompanied by engineering and manufacturing hurdles arising because of product proliferation.

There has been a gradual paradigm shift in the manufacturing strategy in the later part of the nineteenth century. Companies no longer subscribe to the surmise followed by Henry Ford, capturing market share by producing large volumes of a

[1] Assistant Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004, USA

[2] Department of Industrial Engineering, University of Houston, Houston, Texas 77004, USA

A.K. Kamrani and E.S. Abouel Nasr (eds.), *Collaborative Engineering: Theory and Practice.*
DOI: 10.2007/ 978-0-387-47321-5, © Springer 2008

standardized product. Businesses have realized that responding to the market quickly with specific product targeting to niche markets will be the crucial factor that will give them a cutting edge over their competitors. In the automobile industry, this has resulted in a large number of car models that are being offered worldwide. In the USA, the world's largest automotive market, this trend has been particularly dramatic—from 84 models in 1973 to 142 models in 1989, an increase of almost 70%. The volume of production per model sold in the USA has also dropped significantly. During the same period, the average annual sales per model over the lifetime of the product have dropped by 34% from 169,000 units to 112,000 units per model [22].

One of the critical aspects of modern manufacturing is its ability to handle product variety. Production volume is the crucial factor that dictates the extent of product variety because product variety and production quantity are inversely correlated. Thus, job shop production is the most flexible and mass production is the least flexible to handle product variety, the reason being that offering high product variety in mass production increases internal variety costs related to inventory, setup, utilization of equipment, material overhead, floor space, configuration, and customization. Hence at a larger scale of production, modern manufacturing is constantly challenged by its ability to handle high product variety.

11.2 Product Variety

Product variety in a manufacturing setup is the different product designs or product types that are produced in a plant. This is further classified into hard variety and soft variety. Hard product variety occurs when products differ substantially in appearance. In an assembled part, hard variety is characterized by a low proportion of common parts. An example for hard variety would be a car and a truck. Soft product variety occurs when there is a high proportion of common parts and there are only small differences between products. An example for soft variety will be the different car models.

Product variety can be broadly classified into external variety and internal variety [3]. External variety is product variety seen by the customer and internal variety is the product variety experienced inside manufacturing and distribution operations. External variety is further classified into useful variety and useless variety. Useful variety is appreciated by the customer, and useless variety is transparent, is unimportant, or confuses the customer. Internal variety usually takes the form of excessive and unnecessary variety of parts, features, tools, fixtures, raw materials, and processes.

An alternative classification of product variety is fundamental variety, part variety, and peripheral variety [23]. The difference between models and body styles is defined as the fundamental variety. Parts variety includes the combinations of the subsystems such as engine/transmission combinations and the n umber of interior/exterior color combinations. The peripheral variety deals with all the different options available per model.

American and Japanese automobile (hereafter referred to as “auto”) manufacturers have different production strategies to tackle product variety. American car manufacturers believed in the philosophy of mass production, producing high volumes and achieving economies of scale. They always chose to offer minimum fundamental variety to keep the production costs low and offered very high peripheral variety [22]. Japanese auto manufacturers offered more distinct models to choose from and less option combinations, thus increasing the fundamental variety. For instance, in 1982 option combination for Ford Thunderbird was 69,120 models, whereas Honda Accord had only 32 different models [20]. One of the operational reasons for this fundamental difference in product variety was their adversative production strategies. American manufacturers followed the push production system and the Japanese manufacturers followed the pull production system.

In the push production system, batches of materials are forced through different stages of production. The flow of materials is planned and controlled by a series of production schedules. The push system is also called the make-to-stock system. The ability to handle product variety in the push system is governed by the setup time. In most cases, because of the high setup times the resulting batch sizes are large, rendering the push system the least flexible to handle product variety.

In the pull production system, there is an emphasis on reducing inventory at every stage of production. The successive station triggers production at the preceding station which ensures production only if there is downstream demand. The pull system is commonly known as just-in-time (JIT) manufacturing because the product and the corresponding subassemblies are produced as and when required. The JIT philosophy (often coupled with low setup times) facilitates lower work in process (WIP) and is popularly known as the lean manufacturing methodology. The agile nature of the pull manufacturing system offers higher flexibility in manufacturing to accommodate higher product variety. Three big plants in North America and new entrant plants have lower levels of product variety [12].

11.3 Manufacturing Complexity

The extent of product variety handled by any manufacturing firm depends on its volume of production. Low-quantity production usually known as the job shop production or craft production makes small quantities of specialized and customized products using general purpose equipments and flexible layouts. The agility embedded in job shop makes it the most conducive production system to handle high product variety. Medium-quantity production handles hard product variety by batch production, in which batches of materials are pushed through different stages of production and assembly. There is a fixed time for change over from one batch to the other known as the setup change time or the change-over time. Soft product variety in medium production is handled by cellular manufacturing, where every cell specializes in the production of a given set of similar parts or products, according to the principles of group technology. The layout is often known as cellular layout because each cell is designed to produce a limited variety of part configurations.

Manufacturing complexity is defined as a systematic characteristic which integrates several key dimensions of the manufacturing environment, which include size, variety, information, uncertainty, control, cost, and value. Manufacturing complexity is classified into structural (static) complexity and operational (dynamic) complexity [17]. Structural or static complexity is defined as the expected amount of information necessary to describe the state of a system [8, 9, 16]. Production schedule provides the data to calculate the static complexity of the manufacturing system. Static complexity is measured using the entropy equation

$$H_s = -\sum_{i=1}^{m}\sum_{j=1}^{S} p_{ij}\log_2 p_{ij} \tag{11.1}$$

where m is the number of resources, s is the number of scheduled states, and pij is the probability of resource i being in scheduled state j.

Operational or dynamic complexity is defined as the expected amount of information necessary to describe the state of the system deviating from schedule due to uncertainty. The calculation involves the measurement of the difference between actual performance of the system and the expected figures in schedule. Dynamic complexity is given by

$$H_d = -P(\log_2 P) - (1-P)\log_2(1-P) - (1-P)\sum_{i=1}^{m}\sum_{j=1}^{n_s} p_{ij}\log_2 p_{ij} \tag{11.2}$$

where P is the probability of the system being in control, (1 - P) is the probability of the system being out of control, m is the number of resources, n_s is the number of non-scheduled states, and p_{ij} is the probability of resource i being in non-scheduled state j.

A fair estimate of the cost of increased product variety is often difficult to estimate because variety incurs many indirect costs which are not clearly understood and are not easy to capture. Costs that are difficult to determine include raw material inventory, WIP inventory, finished goods inventory, post-sales service inventory, reduction in capacity due to frequent setups, and cost of increased logistics due to added variety [19]

Setup time or the batch size primarily determines the cost of variety in a manufacturing setup. Because of large volumes, mass production has specific machinery that is relatively inflexible for handling product variants. Furthermore, mass production is often characterized by dies that have large setups which encourage higher lot sizes. This forces mass production to have large batch sizes in order to minimize the downtime per product. Consequently, this results in larger WIP, larger floor space, lower quality costs, and lower machinery utilization. WIP inventory costs have a direct relation with respect to the lot size in any manufacturing setup. With higher WIP, the production system drifts toward the push system of production. This is often accompanied by an increase in floor space utilization and an increase in internal transportation costs. Large lot sizes increase the quality costs due to repeating errors, the primary reason being the increase in vulnerability to the unknowingly occurring manufacturing defects. Smaller lot sizes favor lesser part rejection and result in lower quality costs (cost of rejection). Machine utilization increases when the lot size increases because of the economy of scale. Thus, when there is a drift toward increase in product variety in mass production, machine utilization suffers. In traditional mass production, frequent setups to accommodate wide product range also increase the setup cost, labor costs, and downtime. In summary, mass production is least flexible in handling product variety because of operational inefficiencies and cost increases due to product proliferation.

This chapter will present a new methodology for analyzing manufacturing complexity due to increased product variety. A cost model is proposed that captures the increase in inventory and storage cost of the subassemblies due to an increase in product variants. This is accomplished by capturing the product variety and generating a mixed-model assembly sequence that aims to minimize the variation of subassembly inventories of the production span. The mixed-model assembly line is simulated in WITNESS® simulation software to track the inventory levels of the individual subassemblies. The output of the simulation model is used in the cost model to give the daily cost of inventory holding and storage of the different subassemblies.

11.4 Mixed-Model Assembly

The inventory level in a mixed-model assembly line is primarily determined by the sequence of the vehicle models that are assembled on the line. Monden in Toyota Production Systems argues that if products with longer processing times are successively fed into the assembly lines, a delay in model completion will eventually occur [28, 32]. To prevent this from occurring, the processing time at each station must be managed by sequencing models so that, in general, a model with relatively short processing time at a station follows soon after a model with relatively long processing time. Furthermore, the quantity of each part used per unit time must be as near constant as possible because it is crucial for the processes preceding the assembly line supplying components to have a uniform demand [2, 4, 11]. The uniform demand allows the JIT "pull" system to minimize WIP inventories. Other researchers compare the goal-chasing method and the goal-chasing method II developed at Toyota, Miltenburg's algorithm, and time spread methods of sequencing for the assembly line efficiency factors such as work not completed, worker idleness, worker station time, and a measure of variability in uniform component usage [33]. Their study showed that time spread and Miltenburg's algorithms were the most effective overall sequencing procedures. Time spread was preferable if assembly line efficiency was considered, whereas Miltenburg's algorithm performed better if part usage was given primary importance. Aigbedo and Monden experimentally investigated the effect on subassembly usage smoothening when the product usage smoothening is considered together with the subassembly usage smoothening goal for determining the assembly line product sequence [1]. The impetus for their study was Miltenberg's remark that Toyota neglected the product usage smoothening and considered only the part usage smoothening while determining the assembly line product sequence. The results of their study showed that two-level scheduling was computationally faster but in general performed poorer than single-level smoothening. Drexl and Kimms formulated the JIT mixed-model assembly line sequencing problem as an integer program considering both part usage constraints and station load constraints [15]. Their results showed that solving the LP relaxation of the problem by column generation provides tight lower bounds for the optimal objective function value. Garcia and Sabater claimed that mixed-model assembly sequencing considering leveled component consumption, option appearance, and smoothened workload is dynamic in nature because of the availability of the different products to be sequenced [18]. They also propose and test an efficient parametric procedure that adapts to the varying conditions.

Mane et al. tested Toyota's goal-chasing algorithm I and user-defined algorithm on an Australian automotive company [24]. They concluded that the goal-chasing algorithm generated sequences better than the user-defined algorithm although the latter was flexible in accommodating user-defined priorities.

Simulation has been conventionally used to model assembly lines due to stochastic nature of assembly operations. Muralidhar, Swenseth, and Wilson described process times in a JIT environment using truncated normal, gamma, and log-normal distributions and concluded that gamma distribution was the best suited for describing processing times [28]. Mejabi and Wasserman suggested that during JIT implementation some subsystems will continue to retain their push characteristics [26]. They proposed a control paradigm based on the concept of kanban satisfaction that provides a control structure which permits the pulling of material to take place in a JIT environment. They also claimed that of the high-level languages only WITNESS® possessed any real JIT capabilities. Carlson and Yao compared the push and the pull production systems in handling mixed-model assembly operation [10]. Their results showed that the pull production system performed better than the push production system if the assembly line rejects were low. Higher reject rates in JIT assembly lowered the line performance immensely because of the absence of queues between the stations. They also showed that push systems perform significantly better than pull systems if the defect rates were high. Chu and Shih emphasized the use of simulation in analyzing JIT production systems [11]. They proposed simulation as a successful tool in evaluating factors such as measure of company's JIT performance, acceptable inventory level, and use of two-card or single-card kanban. They claimed that many simulations-related statistics were ignored in previous studies and conclusions drawn from previous studies need to be reconfirmed. They also felt that most researchers had a common perception that some experimental factors were more important than the others and the overall behavior of these factors has not been well explained yet. Wang and Xu tested the performance of a hybrid push/pull production using strategy simulation software for flow shop manufacturing [34]. The model simulated the material flow for different production strategies and the simulation results demonstrated that the recommended push/pull strategy was the best for the general mass product manufacturing systems. Baykoc and Erol modeled a multi-item, multistage JIT system in SLAM II, a FORTRAN-based simulation language, and analyzed the effects of increase in the number of kanbans on production performance [5]. Their results showed that the output rate and utilization increase as the number of kanbans increase, but no improvement is observed after two kanbans. Also, increasing the number of kanbans result in a striking increase in waiting times and WIP lengths. Hence, they concluded that the ideal number of kanbans was two for the system considered in their study. Their results also showed that better performance on a mixed-model JIT system depends on reducing or eliminating (if possible) variations related to assembly time, demand arrivals, and balance between stations. Savsar simulated an assembly of printed circuit board and analyzed push and pull production systems [30]. His study shows that the simulation modeling approach can be utilized to determine the minimum kanbans needed to circulate in the system and the WIP buffer levels needed to meet a specified percentage of demand on time in a real assembly line setting. Bukchin studied the throughput of a mixed-model assembly line using six performance measures through simulation [7]. The performance measures include smoothed station, minimum idle-time, station's coefficient of variation, bottleneck, and model variability. Bottleneck measure was the best measure in almost all simulation results followed by model variability and smoothed station. Spedding and Sun used discrete event simulation to evaluate activity-based costs (ABCs) of a manufacturing system [31]. WITNESS® simulation software was used to model a semiautomated printed circuit board assembly line. Under similar conditions, the simulation model gave same estimates as those derived from the IDEF modeling approach. However, simulation models had the advantage of being able to provide greater detail and take into account the intrinsic variation of a dynamic manufacturing system. Akturk and Erhun classified techniques to determine both the design parameters and the kanban sequences for JIT manufacturing [2]. They observed that JIT is based on repetitive manufacturing. Therefore, factors that adversely affect the repetitive nature of the system such as increasing the product variety and decreasing the product standardization reduce the performance of kanban systems. It was also observed that perfectly balanced lines outperform the imbalanced ones even when we vary the number of kanbans at each stage. Taylor compared the potential benefits of three WIP inventory systems: push, pull, and a hybrid push–pull system using SIMFACTORY simulation software [33]. The results showed that the push system averaged the highest inventory level followed by pull and hybrid systems. The push system also had the least profits followed by pull and hybrid production systems. An interesting observation was that although there was a substantial difference between the average inventory levels of pull and hybrid systems the difference in their profits was minimal. Detty and Yingling used discrete event simulation to quantify the benefits of conversion to lean manufacturing [13]. Their results showed an average reduction in waiting time of parts, model change-over times, floor space, and average inventory levels in the lean system as compared to the existing system. A noteworthy result was that the lean system had an 86% reduction in average lead time compared to the traditional manufacturing system. Martinez and Bedia used the modular capabilities of WITNESS® to introduce a modular simulation tool [25]. They built a U-shaped line by integrating a feeder double kanban line module. Their studies demonstrate the use of modular capabilities of WITNESS® in analyzing system configurations and scheduling rules before implementing them. Li used a simulation-based approach to compare push and pull production environments considering the context of JIT implementation [21]. They found that setup reduction effected by cellular manufacturing substantially affected the one-piece production and conveyance in job shop environment.

11.5 Impact of Product Variety on Manufacturing Costs

The ideal product variety to offer and the cost of added product variety have been approached differently by researchers in the past [29]. Malik and Sullivan used mixed integer programming that utilizes ABC information to determine optimal product mix and product cost in a multiproduct manufacturing environment [23]. They showed by an example that with the traditional costing approach, it was possible to arrive at a product mix which may not be achievable with a given capacity of indirect resources. Furthermore, adopting a product mix strategy suggested by traditional costing methods might also increase the overhead costs which are not anticipated in the early stages of planning and costing. MacDuffie, Sethuraman, and Fisher examined the effect of product variety on manufacturing performance [22]. The performance factors include total labor productivity and consumer-perceived product quality. They defined three dimensions of product variety: fundamental, peripheral, and intermediate variety. Their study supports the hypothesis that lean production plants are capable of handling higher levels of product variety with less adverse effect on total labor productivity than are traditional mass production plants. Their study partly explains how the leanest Japanese plants have been able to achieve higher overall performance with much higher levels of parts complexity and option variability. Ishii and Martin introduced the concept of design for variety (DFV) which is a tool that enables product managers to estimate the cost of introducing variety into their product line [19]. They claimed that cost estimates used to determine the profitability of the companies that offered new product offerings did not account for all the costs associated with providing this additional variety. Their model attempts to capture the indirect cost of variety through the measurement of three indices: commonality, differentiation point, and setup cost. DFV methodology was a basic procedure for helping managers and engineers understand the true costs of introducing variety into their product line. Benjaafar, Kim, and Vishwanadham examined the impact of product variety on inventory costs in a production inventory system with finite capacity assuming make-to-stock production, setup times, finite production rate, and stochastic production times [6]. Their results show that inventory costs increase linearly with the number of products. They also show that the rate of increase is sensitive to system parameters including demand and process variability, demand and capacity levels, and setup times. Dobson and Yano formulate the problem of product variety and pricing as a nonlinear program [14]. They assume a manufacturer who has a single machine or production line which is capable of producing a range of potential products. The effect of inventory costs associated with the products is captured by modeling the time between production runs as a decision variable. Their results show that the optimal product mix depends strongly on the production cycle duration. Ozbayrak, Akgun, and Turker estimated the manufacturing costs of an advanced manufacturing system that runs under a material requirement planning (MRP) or JIT system by using activity-based costing [28]. They use simulation as a modeling tool to observe the manufacturing cost behavior under two separate control strategies, the push system and the pull system. Parts are either pushed or pulled and are sequenced according to the four scheduling rules, which are shortest processing time, longest processing time, first come first serve, and slack. They found randomness, buffer capacity, and lead times to be important cost drivers in terms of their effect on WIP and throughput, and that an increase in variation and buffer capacity can result in a build up of WIP inventory and a slight increase in throughput volumes with the expense of considerable increase in manufacturing costs.

11.6 Problem Overview

The research site for the project is the axle assembly operation of a major automobile company that assembles a variety of vehicle models with different combinations of axles and spring coils. Axles and coils are delivered daily based on the scheduled production, and a level of safety stock is always available in the event of any out-of-sequence production. Rear and front axles are installed onto the vehicle chassis on a moving platform at the first two stations and then moved through the line for other tasks such as brake line and spring coil assembly. The following are the different subassembly variants of front/rear axles and front/rear coil springs in the assembly line:

- Nine front axles: 184AP, 184AQ, 187AQ, 187AR, 600AC, 600AD, 601AC, 601A, and 601AE
- Four rear axles: 426AG, 429AG, 430AF, and 433AG
- Seven rear spring coils: 344, 345, 400, 404, 500, 550, and 551
- Seven front spring coils: 262, 263, 264, 265, 267, 268, and 269

Theoretically, there are 1,764 product combinations of front/rear axles and front/rear spring coils possible, but in reality there are only 55 vehicle models allowed because of the design and operational qualifiers. For instance, a heavy duty axle cannot match with a low stiffness spring. A pictorial representation of the possible axle and coil combinations is shown in Fig. 11.1. The assembly line has an operation cost (inventory holding and storage) corresponding to the current inventory level. The problem is to determine the variation in inventory level and the corresponding operational costs if an additional model (product variant) is introduced into the manufacturing system. The study focuses only on the possible vehicle variants due to various axle and spring coil combinations. The impact of product variety due to the engines, transmission, and transfer case is not considered in this study.

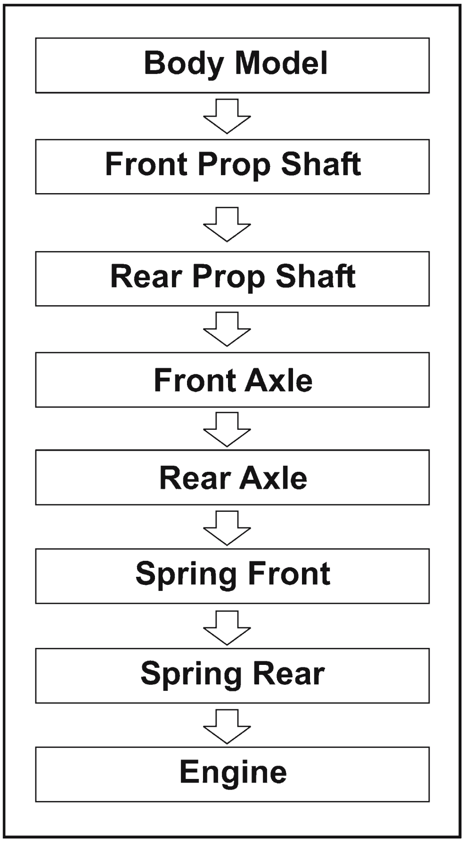


Fig. 11.1 Schematic view of the possible combinations of subassemblies

11.7 Simulation-Based Methodology

Prior to the development of the impact of additional variety on cost, it is necessary to identify the best possible method of part delivery and storage. Three possible scenarios were considered. These are push delivery, sequencing delivery, and in-house sequencing delivery models. Currently, axles are delivered daily based on the scheduled production. Safety stock is always available in case of any out-of-sequence production event. This value is estimated by the axle supplier. Out-of-sequence events could occur because of wrong part sequence, added schedule, and other situations such as breakdown and part damage. Wrong axles are transferred manually and placed in the pull-off bins. In this case, the bins for safety stock are searched until the right axle is located and assembled on the platform. Excess inventory is stored at various locations in the plant. The objective of this phase of the project is to minimize the area that is used for storing the excess inventories and to utilize this area for other value-adding operations. Rear and front axles are installed onto the moving platform at the first two stations and then moved through the line for other tasks such as break line and coil assembly. The approach is to determine the most efficient method of delivery in order to reduce the required storage area. Figure 11.2 illustrates the area where the axles and the coils are assembled into the platform.

Figure 11.3 illustrates the scope of this stage. The implementation steps are as follows:

1. Data collection and verification.
2. Simulation model development and verification.
3. Base model complexity analysis.
   - Part-mix complexity analysis (structural).
   - Manufacturing-mix analysis (dynamic).
4. Delivery/inventory policy.

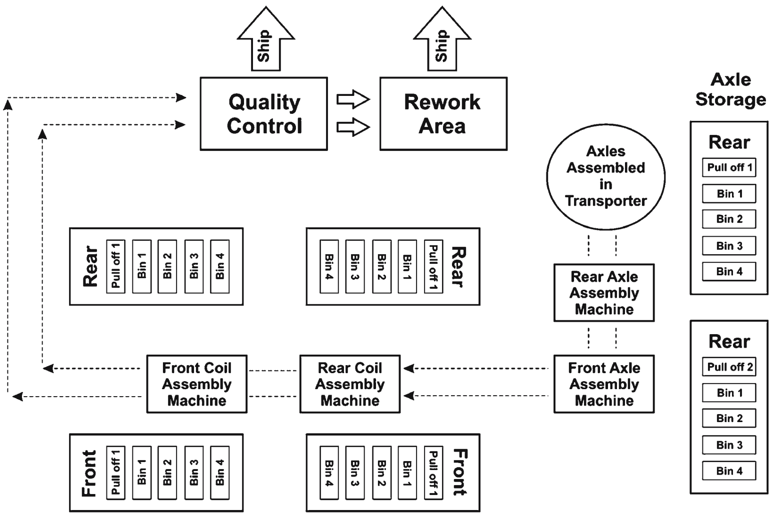


Fig. 11.2 Operational sequence and assembly flow

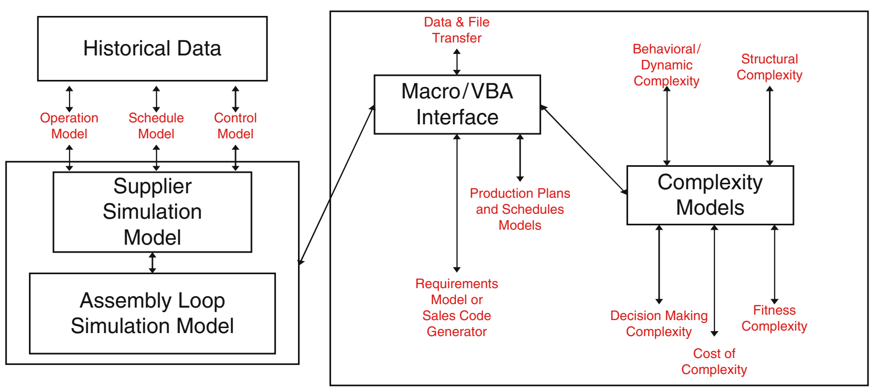


Fig. 11.3 Scope of the proposed system

The required data for implementation is collected using the available historical data for axle operation and daily data collection on the assembly line. Table 11.1 lists the required data for the development of simulation models. Three different simulation models are developed. The first model is based on the current operation of the plant using the sequenced deliverer policy.

Table 11.1 Required data for the simulation model

Required information	Mode type
Production mix data	Front and rear RS spring
	Front and rear LS spring
	Transfer Case T-case
	Rear and front prop
	Rear and front axle
	Engine type
Pull off data	Number of pull off
	Type of axle
	Frequency
	Reason
Delivery information	Delivery schedule
	Number of delivery per day for the axles
	Type of axles
	Number of Hi–Low travel moving/removing per day
	Number of Hi–Low
	Number of delivery per day for springs
	Type of springs
	Number of springs per batch
	Number of Hi–Low travel moving/removing per day
Cycle and stoppages data	Cycle time at each station
	Number of downtime at each station
	Mean time to failure (MTTF)
	Mean time to repair (MTR)
	Reason for failure
	Number of setup
	Setup time at each station
Repair data	Number of repairs due to wrong axles install (done at axle loop)
	Number of repairs due to wrong axles install (not done at the axle loop)
	Repair time for each case
	Delay time before maintenance schedule
	Other repairs due to the process done at the axle loop
	Repair time
	Number of repairs due to wrong spring install (done at the axle loop)
	Number of repairs due to wrong spring install (not done at the axle loop)
	Repair time for each case
	Delay time before maintenance schedule
Facility data	Square footage for the WIP inventory for the springs
	Square footage for the WIP inventory for the axles
	Per square feet cost

The simulation model is consistent with the current mode of operation. As an example, Fig. 11.4 is a comparison chart between the actual production and the simulated one. Based on the results from the statistical analysis, it was concluded that the model is accurate enough in order to continue with the experimentation and complexity analysis.

For this problem, only part-mix complexity (structural) and dynamic complexity (WIP) are measured. For all three models the value of the part-mix complexity is the same ($7.3 \approx 8$ bits), although the dynamic complexity significantly varies from one model to another. Figure 11.5 illustrates an example of the measure of dynamic complexity for the current mode of plant operation (sequenced model).

From the chart, the dynamic measure begins to stabilize at 60%. This is an indicator that the system is undergoing many dynamic changes. This could be the result of changes in production schedules that require axles to be pulled off during assembly. Although the dynamic measure for push and in-house sequence is significantly lower and it is mainly due to the production stoppages (breakdown, repair, etc.). In these

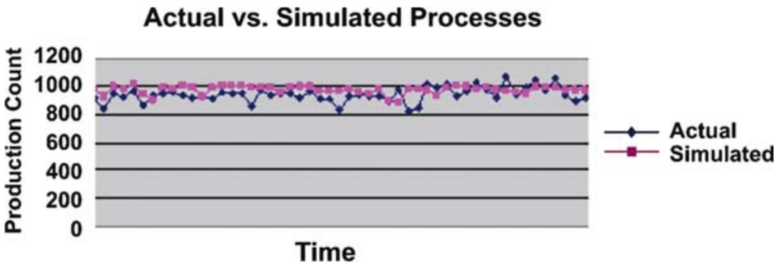


Fig. 11.4 Actual production and simulated production comparison

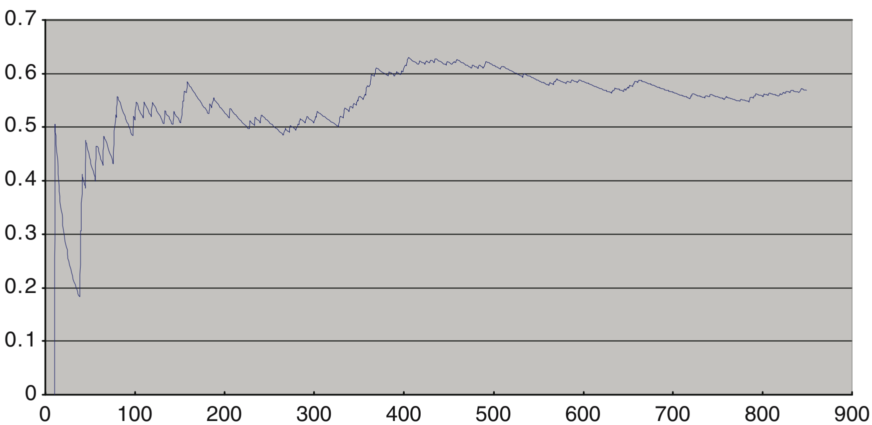


Fig. 11.5 Dynamic measure of complexity for the current scenario for the total built

two cases no parts will be out of sequence. The only impact associated with these two policies is the required inventory. The cost of installation is based on the following:

1. Cost of axle delivered to the assembly plant
2. Cost of storage
3. Cost of Hi–Low operator
4. Cost of pull-off
5. Cost of resequencing
6. Cost of operators
7. Other costs (overtime etc.)

Figure 11.6 illustrates the cost comparisons between all three policies.

Based on the results generated by simulation models, current system layout is well designed to handle the combinations of the vehicle assembly, although the impact of complexity is evident on the cost of material handling and inventory. From Fig. 11.6, it is clear that parts delivered and then sequenced in-house would result in a significant cost saving to the company.
A comprehensive cost model is required to study the impact of the added product variety. This is captured by a sequence-driven simulation model shown in Fig. 11.7.

A three-phase solution methodology is proposed to develop the cost model to quantify the effects of added product variety. In the Sequence Generation phase, a mixed-model assembly sequence using Toyota's goal-chasing algorithm is generated. The goal-chasing algorithm is coded in Visual C++, and it considers only single-level subassembly smoothing. It is assumed that the monthly demands of the vehicle models are known. By knowing subassembly requirements corresponding to the vehicle models, the goal-chasing algorithm generates the mixed-model assembly sequence aiming to minimize the variation in consumption of the subassemblies. In the Assembly Line Simulation phase, the assembly line is simulated using WITNESS® simulation software. The mixed-model assembly sequence

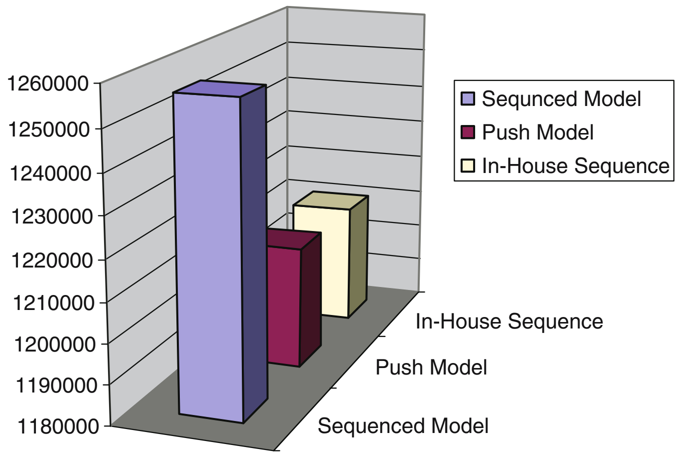


Fig. 11.6 Cost comparisons between different delivery and inventory policies

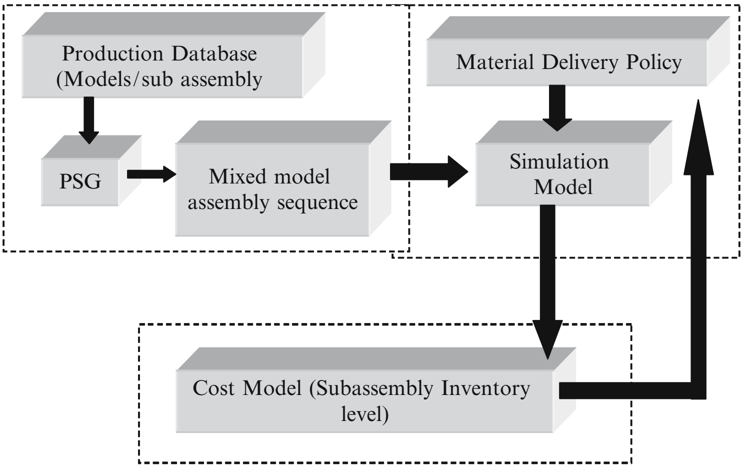


Fig. 11.7 The proposed cost model for added variety analysis

generated by the goal-chasing algorithm is used as input to the simulation model, and the subassemblies are replenished at the beginning of every shift. It is assumed that worker productivity of the assembly line is unaffected by the assembly sequence and there is no shortage of the vehicle models that are scheduled by the goal-chasing algorithm. The assembly sequence follows the JIT methodology, and there is no buffer between the assembly stations [27]. The productivity of the line is primarily governed by the availability of the subassemblies that fit on the vehicle models. The average inventory levels of the subassemblies and the assembly utilization rates are monitored. These parameters from the simulation model are used in the third phase to compute the inventory holding and storage cost of increased product variety in the manufacturing system. Inventory holding cost is a function of the average inventory level and the annual cost of inventory holding. The storage cost is governed by the total number of bins required to store the axles and the spring coils. The number of bins determines the required storage space and the company incurs a storage cost of $40.00/sq ft/day. The central premise of the cost model is that added product variants cause a variation in the mixed-model sequence and consequently the inventory level of the subassemblies. This will be successfully captured by the simulation model and will be reflected in the cost model.

11.8 Experimentations and Results

The cost impact of increased product variety was studied in two production models:

- Daily production sequence shift replenishment model.
- Monthly production sequence hourly replenishment model.

In the daily production sequence model, the production sequence is generated by averaging the monthly demand of the vehicle models and estimating daily demands. The daily sequence generated is repeated every day over the span of 22 consecutive days. The replenishment of all the subassemblies is done at the beginning of every shift. In the monthly production sequence model, the monthly demands of the vehicle models are used to determine the production sequence. The resulting sequence is continuous and the replenishment of most of the subassemblies is done on an hourly basis. Those subassemblies that have a low consumption rate are replenished at the beginning of every shift to reduce the WIP inventory, handling costs, and line stoppages.

11.8.1 Daily Production Sequence with Shift Replenishment

The average inventory levels for subassemblies plotted for four different scenarios are illustrated in Figs. 11.8–11.11. The base scenario (0,0) has 55 product variants with a total production of 1,249 vehicles per day. The second scenario (0,1) has a new product variant (56th vehicle), which uses rear axle 429 AG, front axle 184 AQ, rear spring coil 500, and front spring coil 263. The new production requirement is 1,250 vehicles per day.

The third scenario (1,0) introduces the 57th product variant but removes the 56th product variant. The 57th variant uses rear axle 429 AG, front axle 184 AP, rear axle 500, and front axle 265. The production rate still remains at 1,250/day. The fourth scenario (1,1) includes both the vehicle variants and hence the production rate is 1,251 vehicles per day. The storage cost per day for all the subassemblies is calculated as summarized in Fig. 11.12. The graph shows that there is a steady increase in the storage cost of front and rear axles, but there is only a marginal increase in the storage cost of rear spring coils and no increase in the storage cost of front springs.

The primary reason for the storage cost difference is the storage capacity of the bins. A storage bin can store only 10 axles or 192 spring coils. Therefore, when there is an increase in the average inventory level of the subassemblies, the

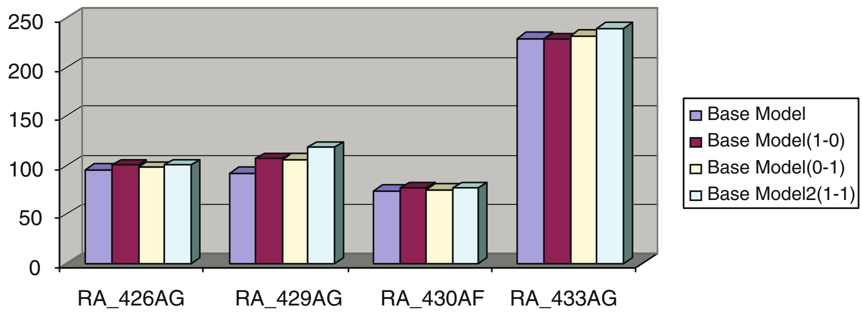


Fig. 11.8 Average daily inventory level of rear axles

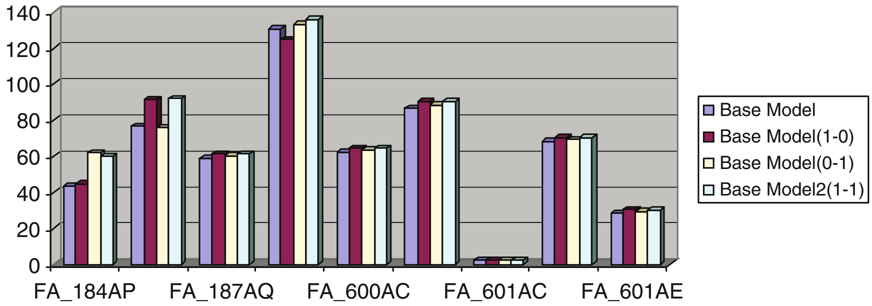


Fig. 11.9 Average daily inventory level of front axles

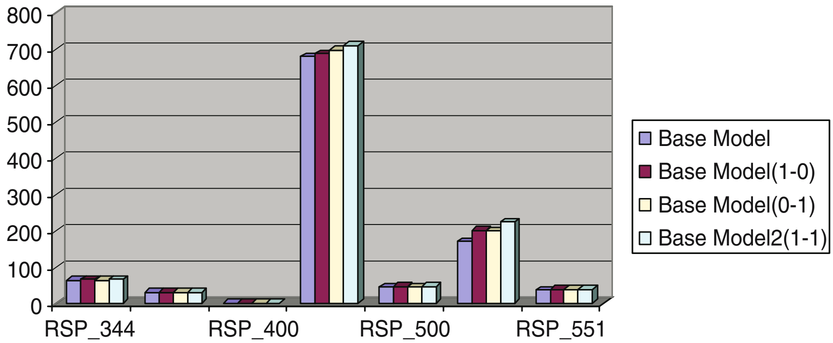


Fig. 11.10 Average daily inventory level of rear spring coils

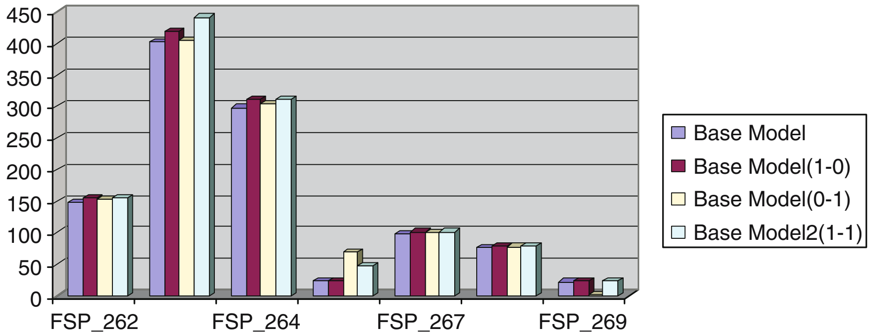


Fig. 11.11 Average daily inventory level of front spring coils

number of bins corresponding to the axles increases drastically, but the number of bins corresponding to the springs increases marginally. This explains the considerable increase in storage cost of the axles compared to that of the springs. Furthermore, the consumption of subassemblies does not have a similar pattern. Among the subassemblies, the consumption rate of the individual models also

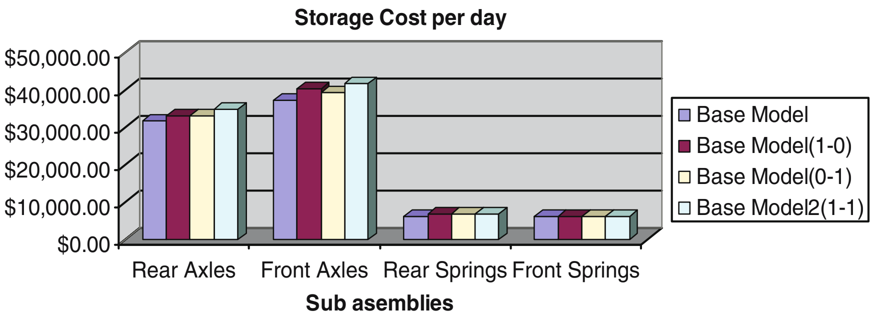


Fig. 11.12 Average daily storage cost for all subassemblies

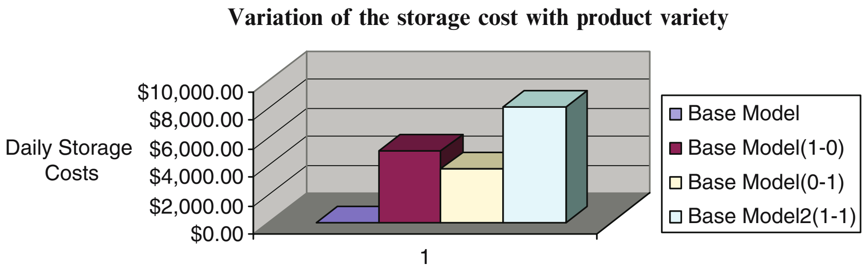


Fig. 11.13 Increase in storage cost with added product variety

makes a difference in the ability to handle product variety. For instance, among the front spring coils, there are a few spring models that are consumed at a very high rate, but the majority of the models have a low rate of consumption. This enables the front spring coils to absorb the added product variety with a corresponding result of no additional storage cost. A gradual increase in the inventory holding cost of the subassemblies is observed with the inclusion of new product variants (Fig. 11.13).

11.8.2 Monthly Production Sequence with Hourly Replenishment

It was observed that with hourly inventory replenishment policy the average inventory level of the high consumption subassemblies was low but that the average inventory level of the low consumption subassemblies was high enough to prevent line stoppages.

Thus, a new delivery policy was proposed and modeled. The high consumption subassemblies are replenished on an hourly basis and the low consumption models are delivered at the beginning of every shift. Front Axle 601 AC; rear spring coils 345, 400, 500, and 551; and front spring coils 265 and 269 are replenished at the

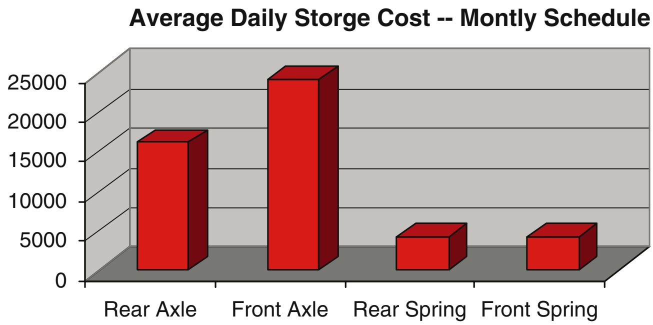


Fig. 11.14 Average daily inventory storage cost for all subassemblies

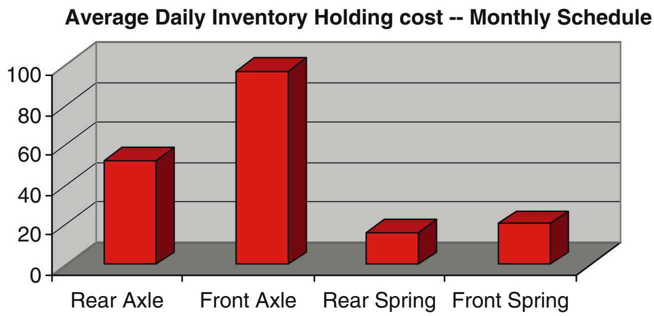


Fig. 11.15 Average daily inventory holding cost for all subassemblies

beginning of every shift and the rest of the subassemblies are replenished hourly. The results are plotted in Figs. 11.14–11.15. The results show that average daily inventory holding cost and storage cost for the monthly schedule hourly replenishment model is less than that for the daily sequence hourly replenishment model. Theoretically, this is accompanied by an increase in material handling cost, but this analysis is not included in the study.

In summary, the proposed methodology provides an insight into the behavior of the manufacturing system by capturing the variation of the inventory holding cost and the storage cost due to added product variety. Although it is expected that the inclusion of a new model will only increase the inventory level (and correspondingly inventory holding and storage costs) of the corresponding subassembly, the results of the analysis show that the inclusion of the new product variant also impacts the inventory level of other subassemblies in the system.

Thus, the final inventory level on the assembly line is determined primarily by the sequence of vehicle assembly and the delivery policy. The simulation model successfully captures these parameters. Hence, the model is an analytical tool for production managers to make informed decisions regarding a new product introduction. The model can also be used to study the inventory holding and storage costs for alternative replenishment policies. The result from the developed methodology proposes that product variety in a mixed-model assembly line can be handled successfully by altering the assembly sequence and the delivery policy. The cost model is generated by capturing the model-mix complexity in a PSG (product structure graph) and generating a schedule to feed the simulation model which runs on the JIT philosophy. The simulation model helps to reveal the impact of added product variety on inventory holding and storage cost. The cost model provides an analytical tool in manufacturing to estimate the projected increase in inventory holding and storage cost and also to study the impact of the manufacturing system on various material handling policies.

11.9 Conclusion

Auto manufacturers worldwide face the constant challenge of striking a balance between product variety and mass production. While increased product variety enables better market coverage by tailoring a product to niche markets, companies struggle with variety to accomplish productivity and quality attained by mass production. Product variety complicates the part supply process. Inventory policies are crucial and decisive factors in improving a company's manufacturing policies. There are conflicting views on holding inventory to strike a balance between the costs associated with holding inventory and not meeting the customer expectations. The reasons for holding inventory include rapid response to customer demands (less order lead time), ordering costs, stock-out costs, and start-up quality costs. The reasons for not holding inventory include inventory carrying costs, loss of system sensitivity, cost of frequent setup changes, and lower return on investment. The proposed method uses a combination of different policies to simplify and improve manufacturing performance.

References

1. Aigbedo, H., and Monden, Y. (1996). A simulation analysis for two-level sequence scheduling for just-in-time (JIT) mixed-model assembly lines. *International Journal of Production Research*, 34(11), 3107–3124.
2. Akturk, M. S., and Erhun, F. (1999). An overview of design and operational issues of kanban systems. *International Journal of Production Research*, 37(17), 3859–3881.
3. Anderson, D. M. (1997). Agile Product Development for Mass Customization. How to Develop and Deliver. Second Edition, Chicago, Irwin Professional.
4. Anwar, F. M., and Nagi, R. (1997). Integrated lot-sizing and scheduling for just-in-time production of complex assemblies with finite setups. *International Journal of Production Research*, 35(5), 1447–1470.

5. Baykoc, O. F., and Erol, S. (1998). Simulation modeling and analysis of a JIT production system. *International Journal of Production Economics*, 55(7), 203–212.
6. Benjaafar, S., Kim, J. S., and Vishwanadham, N. (2004). On the effect of product variety in production-inventory systems. *Annals of Operations Research*, 126(1–4), 71–101.
7. Bukchin, J. (1998). A comparative study of performance measures for throughput of a mixed model assembly line in a JIT environment. *International Journal of Production Research,* 36(10), 2669–2685.
8. Calinescu, A., Efstathiou, J., Bermejo, J., and Schirn, J. (1997). Assessing decision making and process complexity in manufacturing through simulation. Proceedings of the 6th IFAC Symposium on Automated Systems Based on Human Skill, IFAC, Germany, 159–162.
9. Calinescu, A., Efstathiou, J., Huatuco, L. H., and Sivadasan, S. (2002). Classes of complexity in manufacturing. Research Report, Manufacturing Research Group, Department of Engineering Science, University of Oxford.
10. Carlson, G. J., and Yao, A. C. (1992). Mixed model assembly simulation. *International Journal of Production Economics,* 26(4), 161–167.
11. Chu,C. H., and Shih, W. L. (1992). Simulation studies in JIT production. *International Journal of Production Research,* 30(1), 2573–2586.
12. Clair, R., Lafrance, J. C., and Hillebrand, D. (1996), The US Automobile Manufacturing Industry. The US Department of Commerce, Office of Technology Policy.
13. Detty, R. B., and Yingling, J. C. (2000). Quantifying benefits of conversion to lean manufacturing with discrete event simulation: A case study. *International Journal of Production Research,* 38(2), 429–445.
14. Dobson, G., and Yano, C. A. (2002). Product offering, pricing, and make-to-stock/make-to-order decisions with shared capacity. *Production and Operations Management,* 11(3), 293–306.
15. Drexl, A., and Kimms, A. (2001). Sequencing JIT mixed-model assembly lines under station-load and part-usage constraints. *Management Science*, 47(3), 480–491.
16. Frizelle, G. (1996). An entropic measurement of complexity in manufacturing operations. Research Report, Department of Engineering, University of Cambridge, UK.
17. Frizelle, G., and Woodcock, E. (1995). Measuring complexity as an aid to developing operational strategy. *International Journal of Operations and Production Management,* 15(5), 26–39.
18. Garcia-Sabater, J. P. (2001). The problem of JIT dynamic sequencing. A model and a parametric procedure. ORP3, September 26–29.
19. Ishii, K., and Martin, M. V. (1996). Design for variety. A Methodology for understanding the costs of product proliferation. Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, August 18–22.
20. Kamrani, A. K., Adat, A. K., and Rahman, A. (2004). A simulation-based methodology for manufacturing complexity analysis. Proceedings of the National IIE Conference, May 2004.
21. Li, J. W. (2003). Simulation based comparison of push and pull systems in a job-shop environment considering the context of JIT implementation. *International Journal of Production Research,* 41(3), 427–447.
22. MacDuffie, J. P., Sethuraman, K., and Fisher, M. L. (1996). Product variety and manufacturing performance: Evidence from the International Automotive Assembly Plant Study. *Management Science,* 42(3), 350–369.
23. Malik, S. A., and Sullivan, W. G. (1995). Impact of ABC information on product mix and costing decisions. *IEEE Transactions of Engineering Management*, 42(2),171–176.
24. Mane, N., Nahavanadi, S., and Zhang, J. (2002). Sequencing production on an assembly line using goal chasing and user defined algorithm. Proceeding of the 2002 Winter Simulation Conference.
25. Martinez, F. M., and Bedia, L. M. A. (2002). Modular simulation tool for modeling JIT manufacturing. *International Journal of Production Research*, 40(7), 1529–1547.
26. Mejabi, O., and Wasserman, S. (1992). Basic concepts of JIT modeling. *International Journal of Production Research*, 30(1), 141–149.

27. Monden, Y. (1998).Toyota Production Systems—An Integrated Approach to Just in Time. Third Edition, Chapman & Hall.
28. Muralidhar, K., Swenseth, S. R., and Wilson, R. L. (1992). Describing processing time when simulating JIT environments. *International Journal of Production Research,* 30(1), 1–11.
29. Ozbayrak, M., Akgun, M., and Turker, A. K. (2004). Activity based cost estimation in a push/ pull advanced manufacturing system. *Internal Journal of Production Economics,* 87, 49–65.
30. Savsar, M. (1997). Simulation analysis of a pull-push system for an electronic assembly line. *International Journal of Production Economics*, 51, 205–214.
31. Spedding, T. A., and Sun, G. Q. (1999). Application of discrete event simulation to the activity based costing of manufacturing systems. *International Journal of Production Economics,* 58, 289–301.
32. Sumischrast, R. T., Russell, A., and Taylor, B. W. (1992). A comparative analysis of sequencing procedures for mixed-model assembly line in just-in-time production system. *International Journal of Production Research,* 30(1), 199–214.
33. Taylor, L. J., III (1999). A simulation study of WIP inventory drive systems and their effect on financial measurements. *Integrated Manufacturing Systems*, 10(5), 306–305.
34. Wang, D., and Xu, C. G. (1997). Hybrid push/pull production control strategy simulation and its applications. *Production Planning &; Control*, 8(2), 142–151.

Chapter 12
Designing Cellular Manufacturing for Next Generation Production Systems

Ibrahim H. Garbie[1], Hamid R. Parsaei[2], and Herman R. Leep[3]

Abstract Designing cellular manufacturing systems is still under intensive study and has attained significant attention from academicians and practitioners. The major problem in designing cellular manufacturing systems is cell formation. Relevant design objectives, practical issues, and constraints should be taken into consideration. Although there are several cell formation techniques, more work is needed in the areas of the main design objectives, practical issues, and constraints. Over the last three decades, most of the approaches used in cell formation have been based on the machine–part incidence matrix alone and focus only on one or two practical issues sometimes including design objectives and constraints. The practical issues are processing time, alternative routings (process plan), part demand, production volume rate, machine capacity (reliability), and machine capability (flexibility). Hence, solving the cell formation problem is not a simple task, and it must be done concurrently and incrementally. Until now, there has been no practical cell formation approach. This void will lead to the proposal of a new cell formation strategy, which consists of five main phases to improve the quality of solution. In the first phase, a heuristic approach is used to group machines into machine cells based on the similarity coefficient between machines. The second phase uses another heuristic approach to form parts into part families while selecting the best process plans. Initial manufacturing cells are formed in the third phase. In the fourth phase, manufacturing cells are evaluated by measuring the manufacturing cells' performance. Revising the initial manufacturing cells will be included in the fifth phase by considering trade-offs between minimizing the intercellular moves and capital investments, maximizing the efficiency of clustering, and maximizing

[1] Department of Mechanical Engineering, University of Helwan, Helwan, Cairo, Egypt

[2] Department of Industrial Engineering, University of Houston, Houston, 77204 Tx, USA

[3] Department of Industrial Engineering, University of Louisville, Louisville, 40292, Ky USA

A.K. Kamrani, E.S. Abouel Nasr (eds.) *Collaborative Engineering: Theory and Practice,*
doi: 10.2007/ 978-0-387-47321-5,

machine utilization to evaluate the optimal cell design. The proposed strategy was implemented and demonstrated through a numerical example.

12.1 Introduction

Because of an increasingly competitive global market, the need for shorter product life cycles and time to market, and diverse customers, changes in manufacturing systems have been tried to improve the flexibility and productivity of production and manufacturing systems. There are three different types of manufacturing systems: flow shop (mass production) system, batch production system, and job shop production system. The job shop production system is characterized by high flexibility and low production volume and uses general-purpose machines to accommodate fluctuations in part demand and production volume. The flow shop system has less flexibility but more production volume.

Because of the limitations of job shop and flow shop systems, manufacturing systems are often required to be reconfigured to respond to changes in product design, introduction of a new product, and change in product demand and volume. As a result, cellular manufacturing systems (CMS) using group technology (GT) have emerged as promising alternative manufacturing systems.

CMS design is an important manufacturing concept involving the application of GT, and it can be used to divide a manufacturing facility into several groups of manufacturing cells. This approach means that similar parts are grouped into part families and associated machines into machine cells, and that one or more part families can be processed within a single machine cell. The creation of manufacturing cells allows the decomposition of a large manufacturing system into a set of smaller and more manageable subsystems. There are several reasons for establishing CMS. These reasons include reduced work-in-process (WIP) inventories, reduced lead times, reduced lot sizes, reduced interprocess handling costs, better overall control of operations, improved efficiency and flexibility, reduced space, reduced manufacturing costs, improved product design and quality, and reduced setup times. The main disadvantages in cellular manufacturing include high capital investment (machine installation and re-layout), lower utilization, and lack of flexibility in handling demand changes, product mix changes, and product flexibility.

General descriptions of GT and CMS, cell formation techniques, and an extensive review of the various aspects adopted for CMS are discussed carefully in the literature review [21–24, 38, 40, 41, 47, 48, 50, 62, 68, 75, 78, 84, 88, 99, 103, 106, 117, 122–126].

The remainder of this chapter is organized as follows. Section 12.2 reviews the research-related strategy, elements of cell formation, practical issues, and similarity coefficients between machines and between parts. Section 12.3 presents the proposed strategy. A numerical example will be explained in Sect. 12.4. Section 12.5 presents the results and discussion. The conclusions and recommendations for further work are given in Sect. 12.6.

12.2 Literature Review

This section presents a review of research work related to the cell formation approaches proposed in previous works and identifying the strategies which were used. The elements of the cell formation process, including important design objectives, practical issues, and similarity coefficients, are also discussed.

12.2.1 Strategy

The objective of cell formation is to create mutually separable manufacturing cells so that the cells can operate independently with minimum interaction. Cell formation is multiobjective in nature and seeks to satisfy sometimes conflicting goals.

12.2.1.1 Cell Formation Strategy

There are three main solution strategies in cell formation. The first one is a part family grouping strategy; the second one is a machine cell grouping strategy; and the last one is a simultaneous machine–part grouping strategy (Fig. 12.1). The first and second strategies can be considered as sequential strategies, and the third strategy is a simultaneous strategy. The selection of a strategy depends on the designer's philosophy and size of the problem (i.e., number of machines and parts). This chapter concentrates on grouping machines and parts simultaneously.

12.2.1.2 Cell Formation Techniques

In the design of CMS, most cell formation techniques can be separated into two main techniques: mathematical programming and heuristics approaches (Fig. 12.2).

Most cell formation techniques may be considered heuristic techniques, except mathematical programming. *Mathematical programming techniques* can be classified into four categories based on the type of formation, Linear Programming (LP), Integer Programming (IP), Goal Programming (GP), and Dynamic Programming (DP). They are proposed by Vakharia et al. [118], Askin et al. [7], Dahel and

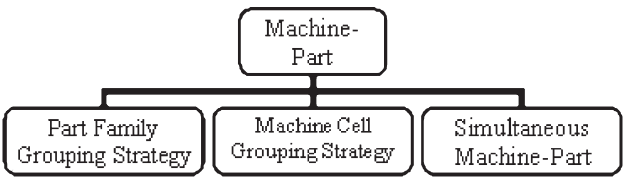


Fig. 12.1 Strategies in cell formation

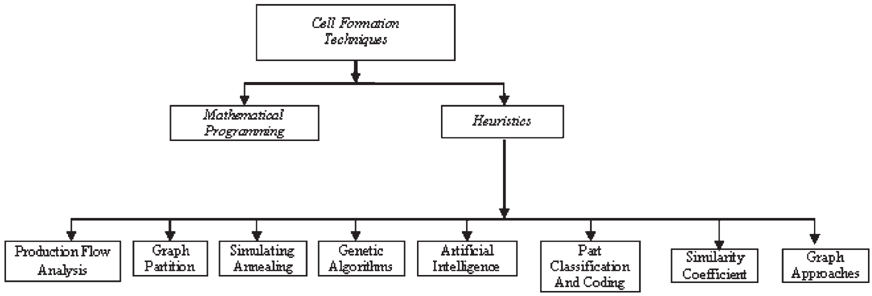


Fig. 12.2 Cell formation techniques

Smith [25], Mungwattana and Shewchuk [71], Abdelmola et al. [2], Sofianopoulou [108], Rajamani et al. [79, 80], Singh et al. [107], Chen [18], Boctor [9], Lozano et al. [56], Akturk and Wilson [3], and Seifoddini [93]. *Heuristic techniques* can be defined as decision procedures or rules that guide the search process toward solving a problem. They are based on the actions selected by the user. The heuristic does not guarantee an optimal solution, but it can generate a good feasible solution in an acceptable time [1]. This means that good heuristic rules lead to good solutions, and bad ones lead to bad solutions [1]. Heuristic techniques can be divided into seven main types: production flow analysis [12], graph partition [67, 70, 132], simulating annealing [8], genetic algorithms [45, 46, 63–65], artificial intelligence [61, 113, 131, 135], part classification and coding [47], and similarity coefficients [72, 97, 98, 121, 129, 130]. Hence, heuristic techniques use some characteristics from other methods to form the part families and machine cells.

In designing CMS, many production and flexibility issues should be included. These issues are operating time, machine capacity (reliability), annual demand per part, production volume, alternative routing (routing flexibility), and machine flexibility. A few cell formation techniques have been developed to incorporate a few of the production and flexibility issues in designing CMS. In this chapter, the proposed heuristic approach based on the two similarity coefficients between machines and between parts will be used in forming part families and grouping machines into machine cells while identifying the best process plan. Improving or revising the manufacturing cells will be considered to achieve a high degree of independence (i.e., minimize the intercellular moves) and to maximize the machine utilization.

12.2.2 Elements of Cell Formation

Cell formation is not a simple task. Vakharia [117] stated that "cell formation should not only be based on one objective; rather it should be a decision based on several objectives which are usually conflicting and thus need to be prioritized."

The cell formation process should take into consideration design objectives, relevant production and flexibility issues, and design constraints (Fig. 12.3).

12.2.2.1 Design Objectives

There are many design objectives that must be achieved in cell formation. These objectives are minimization of throughput times, minimization of setup times, minimization of inventories, minimization of intracell and intercell movements of parts (minimization of material handling costs), minimization of machine relocation costs, minimization of machine load variation, minimization of operating costs, minimization of capital investment, maximization of resource (machine and labor) utilization, and maximization of output. Some of these objectives can be conflicting. These objectives, with regard to a cell formation, can be considered individually or combinatorially [38, 47, 88, 99,106].

12.2.2.2 Design Constraints

There are also some constraints that should be considered while forming a cell such as the following: minimum and/or maximum cell size, minimum and/or maximum number of cells, and maximum number of each machine type.

12.2.2.3 Practical Issues

Several relevant production issues can be incorporated in the process of cell formation such as the following: machine setup time and cost, materials handling costs, production volume and annual demand, machine capacity and machine availability, number of operations per part, operations sequence, processing time per part, machine requirement, alternative routings, and cell layout [21–23, 38, 47, 48, 75, 88, 99, 103, 106, 122, 124].

Production Issues

Several production issues should be incorporated in the design of CMS, such as operating (processing) times, machine capacity, and demand of the part.

- *Operating (Processing) Time*

  Processing time is the time required by a machine to perform an operation on a part type. Normally, setup time and run times are included in processing time.

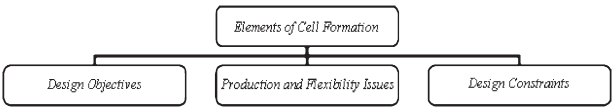


Fig. 12.3 Design cell formation process

The processing time should be provided for every part on corresponding machines in the operation sequence. Processing time is important because it is used to determine resource (machine) capacity requirements. Hence, ignoring the processing times may violate the capacity constraints and thus lead to an infeasible solution [137]. Examples of processing time can be found in several papers [7, 36, 59, 69, 71, 86, 111, 114].

- *Machine Capacity (Reliability)*

  Machine capacity is the amount of time a machine of each type is available for production in each period. When dealing with the maximum possible demand, we need to consider whether the resource capacity is violated or not. In the design of CMS, available capacities of machines need to be sufficient to satisfy the production volume required by parts. Heragu [38] said that machine capacity is more important than the other production factors, and it should be ensured that adequate capacity (in machine hours) is available to process all the parts. Examples of machine capacity are found in papers by Yin and Yasuda [134] and Mungwattana and Shewchuk [71].

- *Demand*

  Demand is the quantity of each part type in the product mix to be produced in each period. The product demand of each part type is expected to vary across the planning horizon. Examples of demand can be found in several papers [4, 7, 71, 86, 119].

Flexibility Issues

In CMS, flexibility can be defined as the ability of the system to adjust its resources to any changes in relevant factors such as product, process, loads, and machine failures [7, 118]. Flexibility could refer to the ability to respond to external disturbances such as volume, mix, and product flexibility, and internal disturbances such as part design and machine flexibility [7, 118]. Although there are at least 50 different terms for various types of manufacturing flexibilities, it is hard to capture the concept [*100*]. There is confusion about the concept of flexibility because of overlapping definitions of different types in different taxonomies in the literature based on the contexts of researchers [33, 54, 116] regarding manufacturing flexibility in general and in CMS to be specific [7, 118]. In discussing flexibility as a management objective, Shewchuk [104], Kumar [32, 35, 51], Gupta et al. [10], Chen and Chung [17], Vakharia et al. [118], and Askin et al. [7] concluded that there was no single measure of flexibility due to its multidimensional definitions and applications. But in practical terms, flexibility has been viewed as a trade-off between efficiency in production and dependability in the marketplace. There exists no rigorous method for identifying the domain of manufacturing flexibility in cellular manufacturing [7, 118].

Flexibility of CMS is currently under intensive study, and the major drawback of most cell formation procedures is the lack of flexibility in designing CMS. Most products (parts) have varying demand and production volume from one period to another, or one or more new products are released to cells every period. The problem gets more complicated when some cells are overutilized, while others are underutilized [111]. This major difficulty occurs when cells stem from unstable machine utilizations due to dynamic and random variations in part demand and/or production volume. The flexibility in a CMS depends on how a machine or group of machines can absorb changes in a given manufacturing environment, changes in demand for products, changes in production volume, changes in costs of operation, changes with the introduction of new parts or products, changes in tooling, and changes in the capacity of machines. Most researchers have dealt with flexibility in general or qualitative terms. Others have attempted to quantify flexibility specifically for manufacturing systems.

Until now, there has been no comprehensive study about flexibility issues in designing CMS. Several types of flexibility issues have been defined by different researchers [87, 100].

- *Machine Flexibility*

  Machine flexibility refers to the capability of machines to perform varying operations without incurring excessive cost from one operation to another. The machine level is fundamental to a manufacturing system, and machine flexibility is a prerequisite for most other flexibilities. Sethi and Sethi [100], Vakharia et al. [118], Askin et al. [7], and Choi and Kim [19] used machine flexibility in cell formation.

- *Routing Flexibility*

  Routing flexibility is the ability of a manufacturing system to produce a product or a part by alternative routes or dynamic assignment of parts to machines with different processing plans. This flexibility will depend on the characteristics of both the product and the equipment. This property is very desirable in situations of equipment breakdown and where uncertainty is prevalent. It has been shown that the flexibility provided by alternative routing creates a very large number of possible routes for each part and is important to consider in forming a configuration of independent cells. Abdelmola [1], Sethi and Sethi [100], Vakharia et al. [118], Askin et al. [7], Albino and Garavelli [6], Dahel and Smith [25, 32, 35], Gupta et al., [71], Seifoddini and Djassemi [95], Sundaram and Doshi [112], Chan [13], Sarker and Xu [89], Wen et al. [127], Ho and Moodie [39], Kannan [49], Albino and Garavelli [5], Drolet et al. [26, 27], Jeon et al. [43–45], Sofianopoulou [108], and Won and Kim [130] proposed routing flexibility in cell formation.

- *Volume Flexibility*

  Volume flexibility of a manufacturing system is its ability to be operated at different overall output levels. This feature will allow the system to deal with volume changes in the current product mix. If the part volume changes, there could be an increase or a decrease in the total number of batches processed in the system. Abdelmola [1], Sethi and Sethi [100], Vakharia et al. [118], Askin et al.

[7], and Shewchuk and Moodie [105] suggested volume flexibility in cell formation. Because of the predominance of routing, machine, and volume flexibilities in the literature review, and because they are the basic components of manufacturing systems, including these characteristics in the design of CMS is very important with the other production issues (operating times, machine capacity, and part demand). In this chapter, the main objectives in the design of CMS is to minimize intercellular movements (minimizing the material handling costs), minimize the number of duplicate machine types, and maximize the machine utilization by incorporating production and flexibility issues, which were explained previously.

12.2.3 Similarity Coefficients

Over the last three decades, many similarity coefficients have been proposed, but a better similarity coefficient between machines and/or parts is still required. Because similarity coefficients can incorporate manufacturing data other than just the binary machine–part incidence matrix, a variety of similarity measures have been defined. The basic idea of CMS design is to take advantage of the similarities in the machines and/or parts. Most clustering algorithms for cell formation rely on the concept of similarity coefficients. This concept is used to quantify the similarity in processing requirements between machines and/or parts, which is then used as the basis for cell formation heuristic methods. The similarity coefficient approaches are a well-known methodology in helping in the design of CMS because they are the most efficient method to group machines and/or parts.

After reviewing 70 articles involving similarity coefficients between machines and between parts [11, 20, 28, 29, 30, 31, 32, 34–37, 39, 42–45, 53, 55, 57, 66, 69, 72, 74, 76, 77, 82, 83, 85, 89, 92–94, 98, 101, 102, 109, 110, 115, 121, 129, 130, 133, 134, 136], one can notice that most similarity coefficients available in the literature on cell formation focus on a single factor and that there are limitations in incorporating various types of production data.

One can also notice that most similarity coefficients, which were used between machines and/or parts, concentrated on data from the machine–part matrix, and few of them took into consideration production data such as production volume, part demand, or processing time. Although a few approaches have been developed to incorporate different factors, there is no comprehensive similarity coefficient between machines and/or parts. The similarity coefficient is flexible in incorporating various types of relevant manufacturing data into the manufacturing cell formation process such as production volume, product demand, process sequence, and machine capacity. It lends itself more easily to computer applications.

Similarity coefficients between parts and/or machines are not absolute, and they still need more attention from researchers. In this chapter, we propose new similarity coefficients between machines and/or parts involving alternative processing routings, processing times, production volumes, annual part demands, machine capacity (reliability), machine flexibility (number of operations done on machine), and maximum number of different operations that can be done on a particular machine.

12.3 The Proposed Cell Formation Strategy

The proposed cell formation strategy will be introduced in five phases. The objective of the first phase is to group machines into machine cells based on the new similarity coefficient between machines. The second phase is used to form parts into part families also based on the new similarity coefficient between parts by identifying the best process plan for each part. The initial formation of manufacturing cells, including machine cells with part families, will be introduced in the third phase. In the fourth phase, the manufacturing cells will be evaluated. In the fifth phase, the initial formation of manufacturing cells will be revised.

Notation

C = number of manufacturing cells.
C_i = capacity of machine i.
C_j = capacity of machine j.
D_k = part demand of part type k per period.
D_p = part demand of part type p per period.
D_q = part demand of part type q per period.
GCI = grouping capability index.
K = subscript of parts (k = 1, …, n).
l = subscript of machines (l = 1, …m).
m = number of machines in the machine–part incidence matrix.
m_c = total number of machines in the cth cell.
MU = machine utilization.
$m_{\max}$ = maximum number of machines into machine cell.
$\mathrm{m}_{\mathrm{X_{prqul}}}$ = number of machines that both part p and part q visit.
n = number of parts in the machine-part incidence matrix.
$N1$ = total number of 1s in the diagonal blocks of the machine–part incidence matrix.
$N2$ = total number of 1s in the off-diagonal blocks of the machine–part incidence matrix.
$N3$ = total number of 1s in the machine–part incidence matrix.
$N4$ = total number of 0s in the diagonal blocks of the machine–part incidence matrix.
NMC = desired number of machine cells.
NPF = desired number of part families.
n_c = total number of parts in the cth cell.
$n_{\min}$ = minimum number of parts in a part family.
$n_{\mathrm{o_i}}$ = number of operations done on machine i.
$n_{\mathrm{o_j}}$ = number of operations done on machine j.
$N_{\mathrm{o_{imax}}}$ = maximum number of operations available on machine i.
N_{ojmax} = maximum number of operations available on machine j.
$n_{X_{ijkr}}$ = number of parts that can visit both machine i and machine j with R process routings.
q = weighting factor ($0 \le q \le 1$) that fixes the relative importance between voids and intercell movements.
r = subscript of alternative routings (r = 1, …, R).
R = number of part routings that can process parts on both machine i and machine j.

$R^{\backslash}$ = number of part routings that can process parts on either machine i or machine j.
S_{ij} = similarity coefficient between machine i and machine j.
$S_{p_r q_u}$ = similarity coefficient between part type p with process plan r and part type q with process plan u.
$t_{ki\mathrm{r}}$ = processing time part k takes on machine i including setup time with process plan r.
$t_{kj\mathrm{r}}$ = processing time part k takes on machine j including setup time with process plan r.
$t_{lp\mathrm{r}}$ = processing time part p takes on machine l with process plan r.
$t_{lq\mathrm{u}}$ = processing time part q takes on machine l with process plan u.
Γ = grouping efficacy.
V_k = production volume rate of part type k per period.
V_p = production volume rate of part type p per period.
V_q = production volume rate of part type q per period.
$X_{ijk\mathrm{r}}$ = 1, if part type k visits both machine i and machine j with process plan r.
$X_{ijk\mathrm{r}}$ = 0, otherwise.
$Y_{ijk\mathrm{r}}$ = 1, if part type k visits either machine i or machine j with process plan r.
$Y_{ijk\mathrm{r}}$ = 0, otherwise.
$X_{p_r q_u l}$ = 1, if part type p with process plan r and part type q with process plan u visit machine l.
$X_{p_r q_u l}$ = 0, otherwise.
$Y_{p_r q_u l}$ = 1, if part type p with process plan r or part type q with process plan r visits machine l.
$Y_{p_r q_u l}$ = 0, otherwise.
η = grouping efficiency.

12.3.1 Phase 1: Grouping Machines into Machine Cells

Machine cells involve the assignment of machines into machine cells based on the new similarity coefficient between two machines, which was described in Section 12.2.3. The procedure to group machines into machine cells will be explained in the following steps:

Step 1: Check the Machine Work Load (MWL) of each machine type capacity ($C_{\_i}$,...,$C_{\_m}$) to produce all parts ($V_{\_1}$,...,$V_{\_n}$) by these machines in the machine–part incidence matrix. The MWL of machine i is based on the production volume rates and processing times of all parts assigned to machine i. The equation for computing the MWL for machine i is shown as follows:

$$\mathrm{MWL}_i = \sum_{k=1}^{n}\left(\sum_{\mathrm{r}_1,\mathrm{r}_2,\mathrm{r}_r \in ki}^{k_{i\mathrm{r}_r}} \max\left(t_{ki_{\mathrm{h}}} V_k + t_{ki_{\mathrm{r}2}} V_k + \cdots + t_{ki_{\mathrm{r}_r}} V_k \right)\right) \tag{12.1}$$

Step 2: Compute the similarity coefficient matrix between all machines according to the following equation:

$$s_{ij} = \frac{\sum_{k=1}^{n x_{ijkr}} \sum_{\substack{r \in ki \\ \text{and} \\ r \in kj}}^{R} \left[\max\left(\frac{t_{kir}}{c_i} \times \frac{n_{o_i}}{N_{o_{i_{max}}}}, \frac{t_{kir}}{c_j} \times \frac{n_{o_j}}{N_{o_{j\,max}}} \right) X_{ijkr} \right] \frac{V_k}{D_k}}{\sum_{k=1}^{n x_{ijkr}} \sum_{\substack{r \in ki \\ \text{and} \\ r \in kj}}^{R} \left[\max\left(\frac{t_{kir}}{c_i} \times \frac{n_{o_i}}{N_{o_{i_{max}}}}, \frac{t_{kir}}{c_j} \times \frac{n_{o_j}}{N_{o_{j\,max}}} \right) X_{ijkr} \right] \frac{V_k}{D_k} + \sum_{k=1+n x_{ijkr}}^{n - n x_{ijkr}} \sum_{\substack{r \in ki \\ \text{or} \\ r \in kj}}^{R'} \left[\max\left(\frac{t_{kir}}{c_i} \times \frac{n_{o_i}}{N_{o_{i_{max}}}} \text{ or } \frac{t_{kir}}{c_j} \times \frac{n_{o_j}}{N_{o_{j\,max}}} \right) Y_{ijkr} \right] \frac{V_k}{D_k}} \quad (12.2)$$

Step 3: Determine the desired number of machines cells (NMC) by the following equation:

$$\text{NMC} \geq \frac{m}{m_{\max}} \quad (12.3)$$

m = number of machines in machine–part incidence matrix.

$m_{\max}$ = maximum number of machines in the machine cell (at least two machines per cell).

Step 4: Select the largest similarity coefficient between machine i and machine (j, …m) from the similarity coefficient matrix in each row directly.

Step 5: Sort the similarity coefficients from the highest to the lowest value and record the values of S_h and the corresponding sets of $m_h\{i,j\}$, where h represents the level of the similarity value.

Step 6: Start forming the first machine cell MC_1 by selecting the highest similarity coefficient valueS_1. Then, this pair of machines $m_1\{i,j\}$will be clustered into the first machine cell.

Step 7: Check the minimum machine cell size constraint (at least two machines per cell).

Step 8: Increase the value of h (h = 2, …, H).

Step 9: If $m_h \cap \text{MC}_1 \neq 0$, then, modify MC_1 by the new $\text{MC}_1 = \text{MC}_1 \cup m_h$. Otherwise, form a new machine cell MC_n (n = 2,...,NMC)

Step 10: If any set m_h intersects two cells MC_I and MC_J, then discard the correspondingS$_h$ and go back to Step 8.

Step 11: Check for the maximum number of machines in a machine cell. If the number of machines in this machine cell does not exceed the desired number of machines, then add to this cell. Otherwise, stop adding to this cell and go back to Step 8.

Step 12: If all the machines have not been assigned to machine cells, go back to Step 8. Otherwise, go to Step 13.

Step 13: If the numbe of machine cells formed exceeds the desired number of machine cellsNMC, join two machine cells into one machine cell.

12.3.2 Phase 2: Assigning Parts to Part Families

Parts are assigned to part families based on the similarity coefficient between two parts, which was described in Sect. 12.3.1. The procedure to group parts into part families by selecting the best alternative routings (process plan) will be explained in the following steps:

Step 1: Compute the similarity coefficient matrix between all parts according to the following equation:

$$S_{p_r q_n} = \frac{\sum_{I=1}^{m_{X_{p_r q_n I}}} \max\left[t_{Ip_r}\left(\frac{V_p}{D_p}\right), t_{Iq_u}\left(\frac{V_p}{D_p}\right)\right] X_{p_r q_u I}}{\sum_{I=1}^{m_{X_{p_r q_n I}}}\left[\max\left[t_{Ip_r}\left(\frac{V_p}{D_p}\right), t_{Iq_u}\left(\frac{V_p}{D_p}\right)\right] X_{p_r q_u I}\right] + \sum_{I=m_{X_{p_r q_n I}}+1}^{m-m_{X_{p_r q_n I}}}\left[t_{Ip_r}\left(\frac{V_p}{D_p}\right) or \left(\frac{V_p}{D_p}\right) t_{Iq_u}\right] Y_{p_r q_u I}} \tag{12.4}$$

Step 2: Determine the desired number of part families (NPF) by the following equation:

$$\text{NPF} \le \frac{n}{n_{\min}} \tag{12.5}$$

n = number of parts in machine–part incidence matrix.
$n_{\min}$ = minimum number of parts in the part family (at least one part per family).

Step 3: Select the largest similarity coefficient between part p and part (q, …, n) from the similarity coefficient matrix with each row identifying the associated process plan.

Step 4: Sort similarity coefficients from the highest to the lowest values, record the values of P_h, and record the corresponding sets of $P_h\{p,q\}$, where h represents the level of the similarity coefficient value including the process plan for each part individually.

Step 5: Start grouping the first part family PF1 by selecting the highest similarity coefficient value$P1$. Then, the pair of parts$P1\{p,q\}$will be grouped into the first part family and the associated process plans will also be determined at the same time.

Step 6: Check for the minimum part family size (at least one part per family).

Step 7: Increase the value of h (h = 2, …, H).

Step 8: If $P_h \cap \text{PF1} \neq \phi$ with the same process plan for any part, then modify PF1 byPF1 = PF1$\cup P_h$. Otherwise, form a new part family PF$_n$ (n = 2,...,NPF).

Step 9: If any set P_h intersects two part families PF$_P$ and PF$_Q$, then discard the correspondingP$_k$ and go to Step 7.

Step 10: Check to determine if some parts have not been assigned to part families, and if so, go to Step 7. Otherwise, stop.

12.3.3 Phase 3: Initial Formation of Manufacturing Cells

Manufacturing cells are formed by assigning part families to machine cells based on the results obtained from grouping machines into machine cells (Phase 1), and parts into part families (Phase 2), and by rearranging the rows and columns of the incidence matrix.

12.3.4 Phase 4: Performance Evaluation

In this phase, the exceptional parts and exceptional machines will be determined. Machine utilization (MU) and efficiency of clustering [grouping efficiency (η), grouping efficacy (Γ), and grouping capability index (GCI)] will also be determined by the following equations:

Machine Utilization (MU) [*58*]:

$$\text{MU} = \frac{N1}{\sum_{c=1}^{c} m_c n_c} \tag{12.6}$$

Grouping Efficiency (η) [*14–16*, *52, 73, 90, 91*]:

$$\eta = q\left[\frac{N1}{\sum_{c=1}^{c} m_c n_c}\right] + (1-q)\left[1 - \frac{N2}{mn - \sum_{c=1}^{c} m_c n_c}\right] \tag{12.7}$$

Grouping Efficacy (Γ) [*52, 90*]:

$$\Gamma = \frac{1 - N2/N3}{1 + N4/N3} \tag{12.8}$$

Grouping Capability Index (GCI) [*96*]:

$$\text{GCI} = (1 - N2/N3) \tag{12.9}$$

12.3.5 Phase 5: Revise or Improve the Initial Manufacturing Cell Formation

Because the varying nature of production activities and the presence of exceptional elements can cause intercellular movements, the extent of cellularization may be less than 100% [125] and around 60% [60]. The ultimate goal of designing a CMS is to convert the entire manufacturing system into independent manufacturing cells. The most common objectives in cell formation are to minimize intercellular movements and maximize machine utilization. Venkataramanaiah and Krishnaiah [120] said that the entire manufacturing system cannot be converted into manufacturing cells, and typically 40–50% of the total production system may need to be separated as an auxiliary cell in order to accommodate exceptional elements.

On the contrary, forcing exceptional elements to go to manufacturing cells reduces the utilization of machines. Three main objectives must be taken into consideration: minimizing the total intercellular movements (minimizing the material handling costs), maximizing the machine utilization and efficiency of clustering, and minimizing the capital investment costs. Therefore, a trade-off between the conflicting objectives is the major problem of interest in the design of manufacturing cells.

Based on these situations, the revised approach takes into consideration the following steps to obtain the best cell formation:

Step 1: Allocate unassigned machines and parts to manufacturing cells.
Step 2: Evaluate intercellular movements for each part, machine utilization, efficiency of clustering, and machine investment individually.
Step 3: Assign identical machines to the manufacturing cells if necessary.
Step 4: Merge two part families or two machine cells if necessary.

Figure 12.4 shows the entire proposed approach to cell formation.

12.4 A Numerical Example

In order to demonstrate the proposed approach, the following numerical example will illustrate the procedure by including the similarity coefficients and the initial formation of manufacturing cells. Machines will be assigned to machine cells and parts will be grouped into part families. The example is composed of ten types of machines and seven types of parts with different process plans. Table 12.1 presents the incidence matrix between machines and parts. Table 12.2 presents the part information including processing sequences and processing times with each process plan, production volume, and product demand. Information about machines available, including the capacity of the machines, number of operations that can be done on each machine, and maximum number of operations available on each machine, is shown in Table 12.3.

12.4.1 Phase 1: Grouping Machines into Machine Cells

Step 1: Check the capacity of each machine type (availability of time per machine) to produce all parts. For machine one, M1, the capacity = 2400 h. The total consumed time taken from M1 will be calculated as follows:

$$\frac{1}{60}\left[\begin{array}{c}2(2000)+\max\left[3.2(2100)+2.5(2100)\right]+1.1(900)\\ +4.83(1800)+2.0(1900)\end{array}\right]=\frac{23{,}784}{60}=396.4\text{h}$$

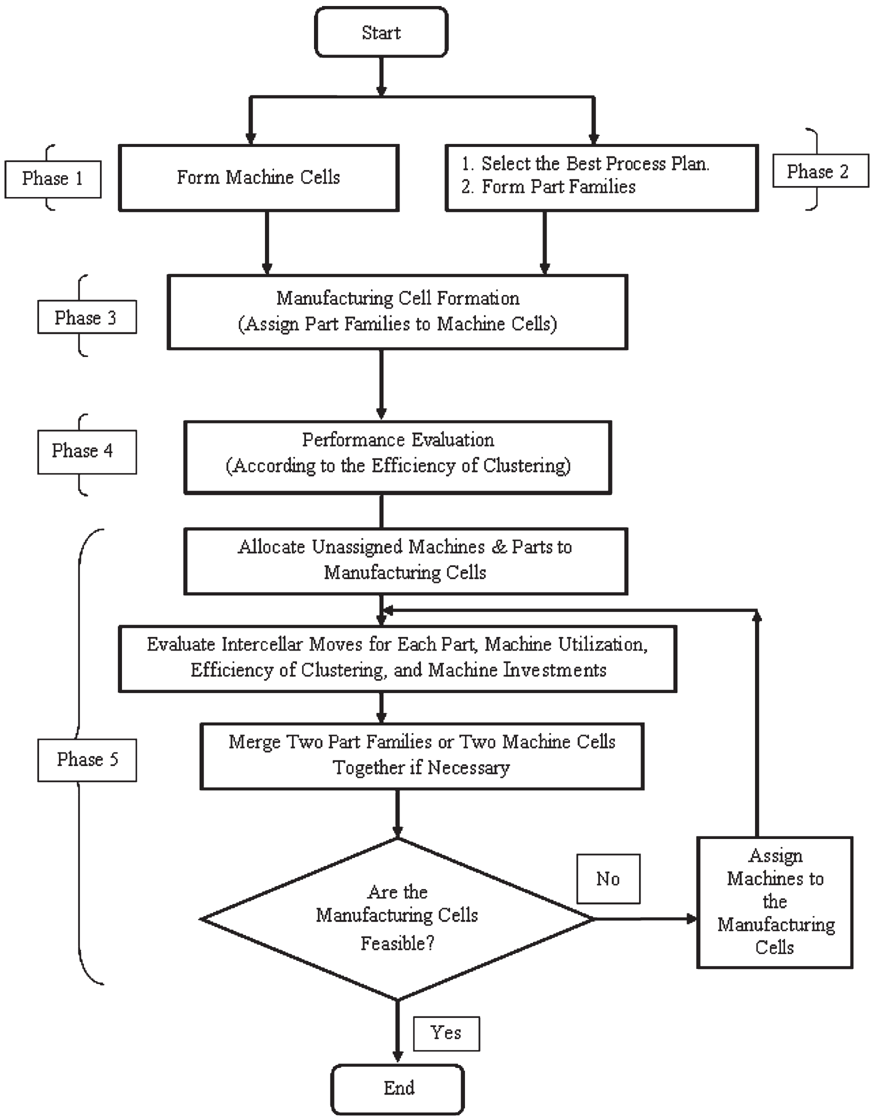


Fig. 12.4 Flow chart of proposed heuristic approach to cell formation

The slack time on machine M1 is 2400 – 396 = 2004 h. So, M1 is okay.
The slack time on machine M2 is 2000 – 412 = 1588 h. So, M2 is okay.
The slack time on machine M3 is 2300 – 439 = 1861 h. So, M3 is okay.
The slack time on machine M4 is 3000 – 509 = 2491 h. So, M4 is okay.
The slack time on machine M5 is 1800 – 144 = 1656 h. So, M5 is okay.
The slack time on machine M6 is 1900 – 453 = 1447 h. So, M6 is okay.
The slack time on machine M7 is 2700 – 229 = 2471 h. So, M7 is okay.

Table 12.1 Incidence matrix between machines and parts

		Parts													
		P1		P2			P3		P4	P5		P6		P7	
		r11	r12	r21	r22	r23	r31	r32	r41	r51	r52	r61	r62	r71	r72
	M1	1	0	1	1	0	1	0	0	0	1	0	1	0	0
	M2	0	1	1	0	1	0	0	1	0	1	1	0	0	1
	M3	0	1	0	1	1	0	1	0	1	0	0	0	1	0
	M4	1	0	1	1	0	1	0	1	0	0	1	0	0	1
Machines	**M5**	1	0	1	0	1	0	1	0	0	0	0	0	0	0
	M6	0	1	1	1	0	0	0	1	1	0	0	0	0	1
	M7	1	0	0	0	1	1	0	0	1	1	1	0	0	0
	M8	0	1	0	0	1	1	1	0	1	0	0	1	1	0
	M9	0	1	0	1	0	1	1	1	0	1	0	1	0	1
	M10	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Table 12.2 Parts information

Part type		Operation sequence	Processing time (minutes)	Production volume	Number of operations per part	Part demand
P1	r11	M1-M4-M5-M7-M10	2.0-3.2-0.9-2.5-0.6	2000	5	1800
	r12	M2-M3-M6-M8-M9	2.7-3.0-4.0-1.35-0.71		5	
	r21	M1-M2-M4-M5-M6-M10	3.0-2.5-0.8-1.1-1.7-2.35	2100	6	2000
P2	r22	M1-M3-M4-M6-M9	2.5-1.8-2.2-3.1-2.11		5	
	r23	M3-M5-M8-M9-M10	2.0-1.2-3.0-1.3-4.4-1.8		6	
	r31	M1-M4-M7-M8-M9	1.1-1.8-2.6-1.5-1.35	900	5	650
P3	r32	M2-M3-M5-M7-M8-M10	3.6-0.6-2.6-0.11-1.93		5	
P4	r41	M2-M4-M6-M9	3.0-3.65-0.5-1.95	2400	4	2000
	r51	M3-M6-M7-M8-M10	4.4-2.83-1.1-2.32-2.0	1800	5	1700
P5	r52	M1-M2-M7-M9	4.83-0.9-0.7-2.28		4	
	r61	M2-M4-M7-M10	1.6-2.1-0.9-1.8	1900	4	1700
P6	r62	M1-M8-M9	2.0-2.3-0.7		3	
	r71	M3-M8-M10	2.0-3.1-3.0	2700	3	2100
P7	r72	M2-M4-M6-M9	0.8-1.9-2.5-4.2		4	

Table 12.3 Machines information

Machine type	Capacity of machine (hours)	Number of operations done on machine (no)	Maximum number of operations available on machine (Nmax)
1	2400	6	6
2	2000	7	7
3	2300	6	6
4	3000	7	10
5	1800	4	4
6	1900	6	9
7	2700	6	8
8	1300	7	10
9	2500	8	9
10	2100	7	10

The slack time on machine M8 is 1300 – 440 = 860 h. So, M8 is okay.
The slack time on machine M9 is 2500 – 475 = 2025 h. So, M9 is okay.
The slack time on machine M10 is 2100 – 383 = 1716 h. So, M10 is okay.
The capacities of all machines are satisfactory at all production volumes for all parts.

Step 2: Compute the similarity coefficient matrix between all machines according to the similarity coefficient equation. The similarity coefficient between machines has been coded in the C programming language and executed on a Pentium IV processor. For example, the similarity coefficient between machine 1 and machine 2 will be explained as the following and the result for similarity coefficients between machines is illustrated in Table 12.4.

$$s_{12} = \frac{\sum_{k=1}^{2}\left[\sum_{\substack{r\in 21\\ r\in 22}}^{1} \max\left(\frac{3.0}{2400}\times\frac{6}{6}, \frac{2.5}{2000}\times\frac{7}{7}\right)\frac{2100}{2000} + \sum_{\substack{r\in 51\\ r\in 52}}^{1} \max\left(\frac{4.83}{2400}\times\frac{6}{6}, \frac{0.9}{2000}\times\frac{7}{7}\right)\frac{1800}{1700}\right]}{\begin{array}{c}\sum_{k=1}^{2}\left[\sum_{\substack{r\in 21\\ r\in 22}}^{1} \max\left(\frac{3.0}{2400}\times\frac{6}{6}, \frac{2.5}{2000}\times\frac{7}{7}\right)\frac{2100}{2000} + \sum_{\substack{r\in 51\\ r\in 52}}^{1} \max\left(\frac{4.83}{2400}\times\frac{6}{6}, \frac{0.9}{2000}\times\frac{7}{7}\right)\frac{1800}{1700}\right] + \sum_{k=3}^{7-31}\{\sum_{r\in 11}^{1}\left(\frac{2.0}{2400}\times\frac{6}{6}\right)\frac{2100}{1800} + \sum_{r\in 12}^{1}\left(\frac{2.7}{2000}\times\frac{7}{7}\right)\frac{2100}{2000} + \sum_{r\in 21}^{1}\left(\frac{2.5}{2400}\times\frac{6}{6}\right)\frac{2100}{2000} \\ + \sum_{r\in 22}^{1}\left(\frac{2.0}{2000}\times\frac{7}{7}\right) + \sum_{r\in 31}^{1}\left(\frac{1.1}{2400}\times\frac{6}{6}\right)\frac{900}{650} + 0 + \sum_{r\in 42}^{1}\left(\frac{3.0}{2000}\times\frac{7}{7}\right)\frac{2400}{2000} + 0 + \sum_{r\in 61}^{1}\left(\frac{2.0}{2400}\times\frac{6}{6}\right)\frac{1900}{1700} + \sum_{r\in 62}^{1}\left(\frac{1.6}{2000}\times\frac{7}{7}\right)\frac{1900}{1700} + 0 + \sum_{r\in 72}^{1}\left(\frac{0.8}{2000}\times\frac{7}{7}\right)\frac{2700}{2100}\end{array}} = 0.1856 \quad (12.10)$$

Step 3: Determine the desired number of machine cells, NMC. The maximum number of machines assigned to cells ranged from 3 to 7 machines [128] and from 5 to 10 machines [81]. Four machines per cell are recommended for easy management and control.

$\text{NMC} \le \frac{10}{4} \ge 2.5$. Therefore, the number of machine cells can start with three cells. Three machine cells will be chosen.

Step 4: Select the largest similarity coefficient between machine *i* and machine (*j*, …, m) from Table 12.4 as follows:

$m_1 - m_7$	0.5072
$m_2 - m_9$	0.5466
$m_3 - m_8$	0.7618
$m_4 - m_7$	0.5097
$m_5 - m_{10}$	0.6015
$m_6 - m_9$	0.6068
$m_7 - m_{10}$	0.4381
$m_8 - m_{10}$	0.6110
$m_9 - m_{10}$	0.0821

Table 12.4 Similarity coefficients between machines

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0.0000	0.1856	0.0739	0.4613	0.2422	0.2219	0.5072	0.1501	0.4751	0.1996
M2		0.0000	0.1816	0.4218	0.3305	0.5031	0.2510	0.2560	0.5466	0.3057
M3			0.0000	0.0698	0.3928	0.4293	0.2458	0.7618	0.3653	0.6190
M4				0.0000	0.2308	0.3287	0.5097	0.0779	0.3326	0.3339
M5					0.0000	0.0751	0.4640	0.3568	0.0511	0.6015
M6						0.0000	0.1312	0.1909	0.6068	0.1942
M7							0.0000	0.3954	0.2516	0.4381
M8								0.0000	0.3363	0.6110
M9									0.0000	0.0821
M10										0.0000

Step 5: Sort the similarity coefficients from the highest to the lowest value and record the values of S_h and the corresponding sets of $m_h\{ i,j\}$

H	$m_h\{ i,j\backslash\}$	S_h
1	$m_3 - m_8$	0.7618
2	$m_8 - m_{10}$	0.6110
3	$m_6 - m_9$	0.6068
4	$m_5 - m_{-10}$	0.6015
5	$m_2 - m_9$	0.5466
6	$m_4 - m_7$	0.5097
7	$m_1 - m_7$	0.5072
8	$m_7 - m_{10}$	0.4381

Step 6: For S_1 = 0.7618 (between Machine 3 and Machine 8). Then, {MC_1 = **{3, 8}**.

Step 7: Check the minimum machine cell size constraint (at least two machines per cell).

Step 8: S_2 = 0.6110 (between Machine 8 and Machine 10),m_2 = {8, 10}.

Step 9: There is an intersection between Machine 8 and MC_1. The new machine cell is $MC_1 \cup_2 m_2$. Then, the revised machine cell MC_1 = **{3, 8, 10}**. S_3 = 0.6068 (between Machine 6 and Machine 9), m_3 = {6, 9}, and $MC_1 \cap_3 m_3 = 0$. S_3 does not intersect with MC_1. Then, form a new machine cell MC_2 = **{6, 9}**. S_4 = 0.6015 (between Machine 5 and Machine10), m_4 = {5, 10}.

There is an intersection between Machine 10 and MC_1, but there is no intersection with MC_2. The new machine cell is $MC_1 U m_4$. Then, the revised machine cell MC_1 = **{3, 5, 8, 10}**.

Step 10: Check for the maximum number of machines in a machine cell. Machine cell 1 contains four machines. Therefore, no more machines are added to {MC_1. S_5 = 0.5466 (between Machine 2 and Machine 9), m_5 = {2, 9}.There is an intersection between Machine 9 and MC_2, but there is no intersection with MC_1. The new machine cell is {$MC_2 U m_5$. Then, the revised machine cell MC_2 = **{2, 6, 9}**, S_6 = 0.5092 (between Machine 4 and Machine 7), m_6 = {4, 7}, $MC_1 \cap_{m6} = 0$, and $MC_2 \cap_{m6} = 0$. There is no intersection between Machine 4 and Machine 7 with either MC1 orMC_2. Then, form a new machine cell MC_3 = **{4, 7}**, S_7 = 0.5072 (between Machine 1 and Machine 7) m_7 = {1, 7}. There is an intersection between Machine 7 and MC_3, but there is no intersection with {MC_1 or MC_2. The new machine cell is $MC_3 \cup_{m7}$. Then, the revised machine cell MC_3 = **{1, 4, 7}**.

Step 11: All the machines have been assigned to machine cells. Stop.
Machine Cells are as follows:
MC_1 = **{3, 5, 8, and 10}**
MC_2 = **{2, 6, and 9}**
{MC_3 = **{1, 4, and 7}**

12.4.2 Phase 2: Grouping Parts into Part Families

Step 1: Compute the similarity coefficients matrix between all parts with all different process plans. The results of all similarity coefficients between parts are illustrated in Table 12.5.

Step 2: Determine the desired number of part families.

$$\mathrm{NPF} \le \frac{7}{1} \le 7$$

Then, the number of part families may range from 7 to 1 (7, 6, 5, 4, 3, 2, and 1). Three part families will be chosen.

Step 3: Select the largest similarity coefficients between part *p* and part (*q*, …, *n*) from Table 12.3, including the possible process plans in each level (row) as follows:

P1-P3 (r11-u31)	0.6255
P1-P7 (r12-u72)	0.6383
P2-P4 (r21-u41)	0.5172
P2-P7 (r22-u72)	0.6642
P2-P3 (r23-u32)	0.8100
P3-P5 (r31-u52)	0.6683
P3-P7 (r32-u71)	0.9288
P4-P7 (r41-u72)	1.0000
P5-P6 (r52-u62)	0.6383
P5-P7 (r51-u71)	0.7502
P6-P7 (r61-u72)	0.2667
P6-P7 (r62-u71)	0.3559

Step 4: Sort the similarity coefficients from the highest to the lowest value and record the values of P_h and the corresponding set P_h $\{P_r, q_u\}$

H	P_h *(p,q)*	P_h
1	P4-P7 (r41-u72)	1.0000
2	P3-P7 (r32-u71)	0.9288
3	P2-P3 (r23-u32)	0.8100
4	P5-P7 (r51-u71)	0.7502
5	P3-P5 (r31-u52)	0.6683
6	P2-P7 (r22-u72)	0.6642
7	P1-P7 (r12-u72)	0.6383
8	P5-P6 (r52-u62)	0.6383
9	P1-P3 (r11-u31)	0.6255
10	P2-P4 (r21-u41)	0.5172
11	P6-P7 (r62-u71)	0.3559
12	P6-P7 (r61-u72)	0.2667

Table 12.5 Similarity coeficients between paris

		P1		P2			P3		P4	P5		P6		P7	
		11	12	21	22	23	31	32	41	51	52	61	62	71	72
P1	11	0.0000	0.0000	0.5896	0.3436	0.3623	0.6255	0.1751	0.2477	0.2247	0.4788	0.6248	0.1636	0.1931	0.1789
	12		0.0000	0.3999	0.5143	0.4848	0.1766	0.4612	0.5298	0.6204	0.2634	0.1627	0.2051	0.3770	0.6383
	21			0.0000	0.4570	0.3876	0.2658	0.1744	0.5172	0.2539	0.4514	0.5118	0.2048	0.1931	0.4049
P2	22				0.0000	0.0743	0.4098	0.3201	0.5483	0.0384	0.4341	0.1350	0.3256	0.1235	0.6642
	23					0.0000	0.3652	0.8100	0.1545	0.6074	0.1581	0.3200	0.2654	0.6255	0.0913
P3	31						0.0000	0.2535	0.3689	0.2789	0.6683	0.3908	0.5228	0.2002	0.4066
	32							0.0000	0.1012	0.6685	0.1132	0.1537	0.2901	0.9288	0.2233
P4	41								0.0000	0.1263	0.3500	0.5686	0.1478	0.0000	1.0000
P5	51									0.0000	0.0532	0.1872	0.1476	0.7502	0.1345
	52										0.0000	0.1903	0.6383	0.0000	0.3556
P6	61											0.0000	0.0000	0.2559	0.2667
	62												0.0000	0.3559	0.3196
P7	71													0.0000	0.0000
	72														0.0000

Step 5: P_1 = 1.000 (between part 4 and part 7 with r41 and u72). Then, group the first part family PF1 **= {4, 7}**, with process plan 1 for part 4 and process plan 2 for part 7.

Step 6: Check the number of parts in the first part family after grouping the first part family.

Step 7: P_2 = 0.9288 (between part 3 and part 7 with r32 and u71). Then, P_2 (3,7) = { 3,7}.

Step 8: Part 7 was already assigned to PF1 with process plan 2, and $P_2 \cap PF1 = 0$ So, it is difficult to group part 3 with PF1.

Step 9: There is no intersection with PF1. So, we discard P_2, and go back to Step 7. P_3 = 0.8100 (between part 2 and part 3 with r23 and u32), P_3 (2,3) = {2, 3}. There is no intersection between $PF1 \cap P_3$. Then, form a new part family PF2 = **{2, 3**} with process plan 3 for part 2 and process plan 2 for part 3.

Step 10: Check if all parts have been assigned to part families. If not, go back to Step 7. P_4 = 0.7502 (between part 5 and part 7 with r51 and u71). P_4 (5,7) = { 5,7} Part 7 was assigned to {PF1}} with process plan 2, and there is no intersection with {PF1}} and {PF2}}. So, we discard P_4, and go back to Step 7. P_5 = 0.6683 (between part 3 and part 5 with r31 and u52). P_5 (3,5) = {3,5} Part 3 was assigned to {PF2}} with process plan 2, and there is no intersection with PF2. So, we discard P_5. P_6 = 0.6642 (between part 2 and part 7 with r22 and u72). P_6 (2,7) = {2,7} Part 2 and part 7 were assigned to PF2 and PF1, respectively, with different process plans. There is no intersection with PF1 and PF2. So, we discard P_6. P_7 = 0.6383 (between part 1 and part 7 with r12 and u72). P_7 (1,7) = {1,7} Part 7 was assigned toPF1 with process plan 2, and $P_7 \cap PF1 \neq 0$. Then, the new PF1 = **{1, 4, 7}**, with process plan 2 for part 1. Check if all parts have been assigned to part families. If not, go back to Step 7. P_8 = 0.6383 (between part 5 and part 6 with r52 and u62). P_8 (5,6) = {5, 6} There is no intersection between $PF1 \cap P_8$ and $PF2 \cap P_8$. Then, form a new part family PF3 = **{5, 6**}, with process plan 2 for parts 5 and 6. Check if all parts have been assigned to part families. If so, stop.

The best process plans are as follows:

Part Number	Process Plan
1	2

12.4.3 Phase 3: Initial Formation of Manufacturing Cells

Forming manufacturing cells by assigning part families to machine cells is based on the results obtained from grouping machines into machine cells and parts into part families (PF). This grouping will be done by rearranging the rows and the columns of the incidence matrix. First, arrange the part families, and then, rearrange the machine cells. Figures 12.5 and 12.6 show these formations.

The manufacturing cells are as follows:

Manufacturing Cell 1 consists of PF1 = {1, 4, 7) and MC1 = {2, 6, 9}

Manufacturing Cell 2 consists of PF2 = {2, 3) and MC2 = {3, 5, 8, 10}

Manufacturing Cell 3 consists of PF3 = {5, 6) and MC3 = {1, 4, 7}

Notice in Fig. 12.6 that there are exceptional elements (parts) and bottleneck machines. Some parts need to be processed in other machine cells in addition to their machine cells. For example, part 1 needs to go to machine cell 2; parts 4 and 7 need to go to machine cell 3; and parts 3 and 5 need to go to machine cell 1. Also, part 2 needs to go to machine cell 1 and machine cell 3, and part 6 needs to go to machine cell 1 and machine cell 2 to complete their operations.

	PF1			PF2		PF3	
	1	4	7	2	3	5	6
	r=2	r=1	r=2	r=3	r=2	r= 2	r= 2
Machine Type 1						1	1
2	1	1	1	1		1	
3	1			1	1		
4		1	1				
5				1	1		
6	1	1	1				
7				1		1	
8	1			1	1		1
9	1	1	1		1	1	1
10				1	1		1

Fig. 12.5 Part families' arrangement

		PF1			PF2		PF3	
		1	4	7	2	3	5	6
MC3	1						1	1
	4		1	1				
	7				1		1	
MC2	3	1			1	1		
	5				1	1		
	8	1			1	1		1
	10				1	1		
MC1	2	1	1	1	1		1	
	6	1	1	1				
	9	1	1	1		1	1	1

Machine Utilization =87.00 %

Efficiency of Clustering:
$\eta = 81.78\,\%$,
$\Gamma = 62.50\,\%$,
GCI = 64.52 %,

Fig. 12.6 Initial formation of manufacturing cells

12.4.4 Phase 4: Performance Evaluation

The values of machine utilization (MU) and efficiency of clustering [grouping efficiency (η), grouping efficacy (Γ), and grouping capability index (GCI)] are as follows:

MU = 87.00%, η = 81.78%, Γ= 62.50%, and GCI = 64.52%.

Although there are seven exceptional parts and six bottleneck machines, the machine utilization is equal to 87.00% and grouping efficiency is equal to 81.78%. These results indicate that the system needs some duplicate machine types to minimize or eliminate intercellular movements.

12.4.5 Phase 5: Revise or Improve the Initial Manufacturing Cell Formation

In order to improve the system and solve these problems, there are many procedures that can be taken into consideration:

Step 1: Allocate unassigned machines and/or parts to a manufacturing cell. Machine 4 is assigned to manufacturing cell 3, but there are no parts in part family 3 (PF3) that needs to go to machine 4. Therefore, machine 4 can be assigned to manufacturing cell 1 which has to process parts 4 and 7. Then, the new manufacturing cells without any additional machines will be shown in Fig. 12.7.

Step 2: Evaluate intercellular moves for each part, machine utilization, and machine investment individually. From Fig. 12.7, the machine utilization is equal to 91.66%, and grouping efficiency is equal to 86.45% with five exceptional parts (parts 1, 2, 3, 5, and 6) and five bottleneck machines (machines 7, 3, 8, 2, and 9). The machine investment is still equal to the total sum of machines (machines 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10).

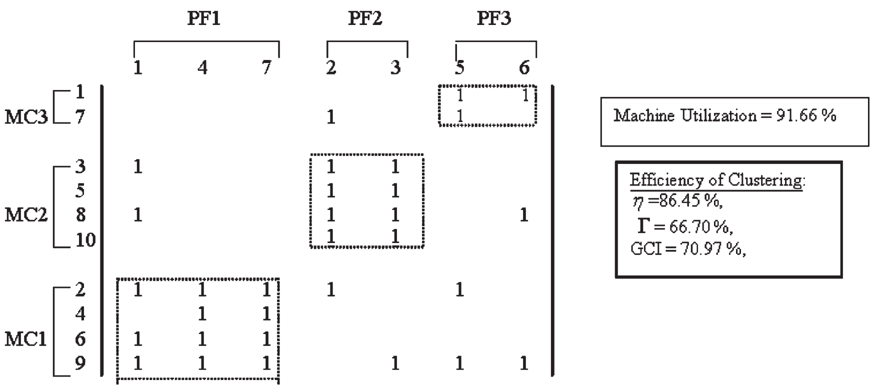


Fig. 12.7 Revised formation of manufacturing cells

Step 3: Add duplicate machines to the manufacturing cells if necessary. Add a duplicate of machine 9 to manufacturing cell 3 to reduce the intercellular moves of parts 5 and 6 to manufacturing cell 1 (Fig. 12.8). The machine utilization is 92.30%, but the number of exceptional parts and bottleneck machines is still the same. The investment in machines is increased by one machine (machine 9). Then, the machine investment is equal to the total sum of machines (machines 1, 2, 3, 4, 5, 6, 7, 8, 9(2), and 10). Steps 2 and 3 are repeated until all the exceptional parts and bottleneck machines are removed from the matrix and are calculated every time with identifying the machine utilization, efficiency of clustering, and machine investments (Fig. 12.9– Fig. 12.11). All these results are shown in Table 12.6.

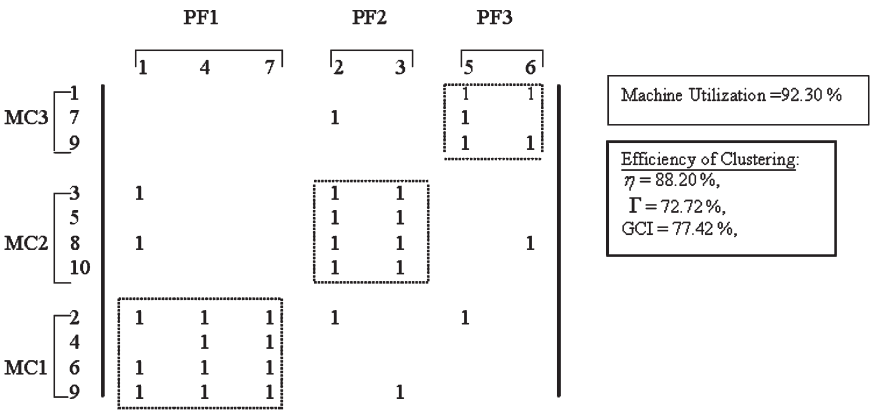


Fig. 12.8 Addition of a duplicate machine 9 to manufacturing cell 3

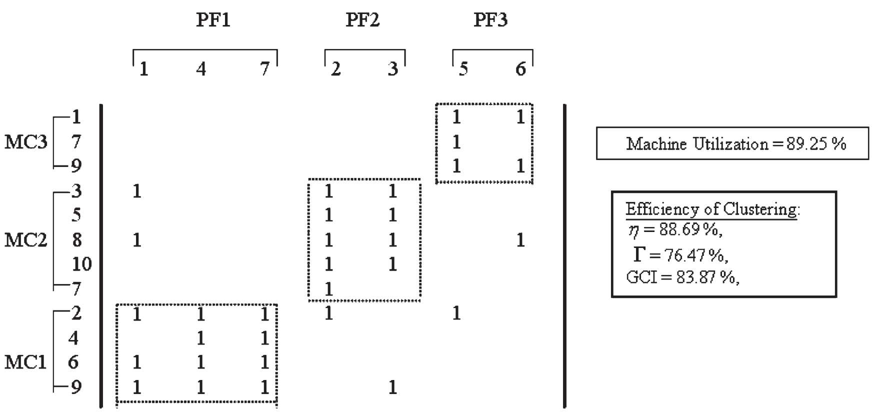


Fig. 12.9 Addition of a duplicate machine 7 to manufacturing cell 2

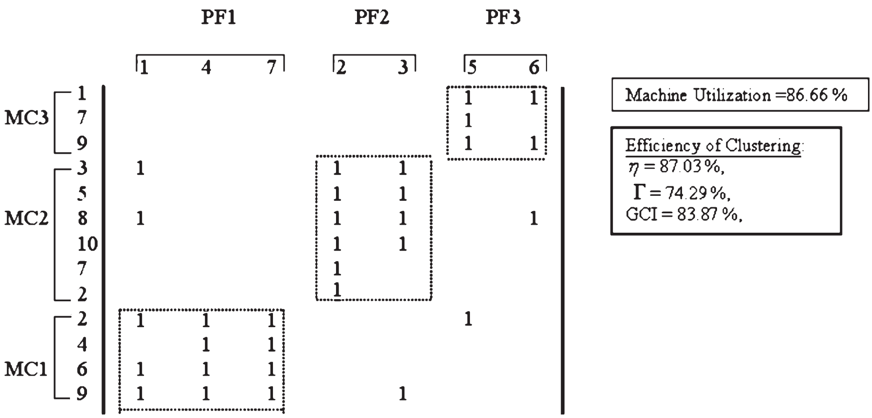


Fig. 12.10 Addition of a duplicate machine 2 to manufacturing cell 2

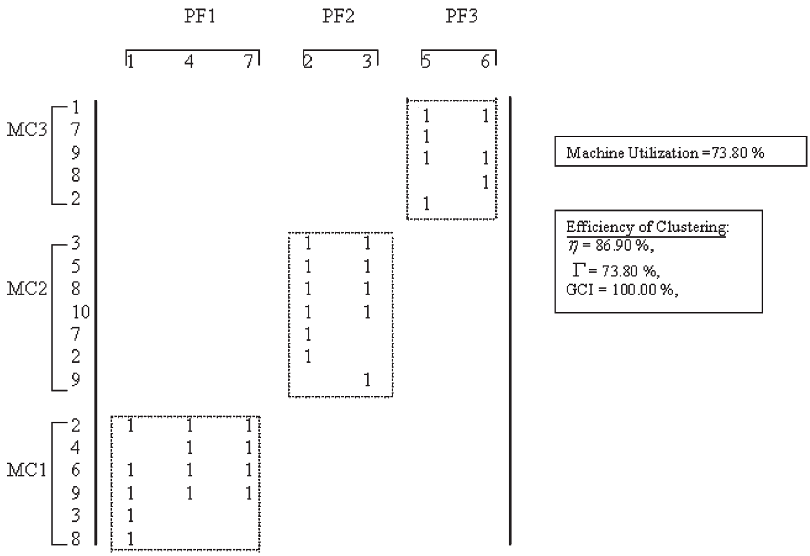


Fig. 12.11 Addition of five more duplicate machines

12.5 Results And Discussion

It should be noted in Table 12.6 for the sixth level that machines 3 and 7 are duplicated twice; machines 2, 8, and 9 are duplicated three times; and no exceptional parts and bottleneck machines exist in the cell formation. It can also be noted for the sixth level that the number of duplicated machines increased to eight machines; machine utilization and grouping efficacy decreased to 73.80%;

Table 12.6 Trade-off between exceptional parts, machine utilization, efficiency of clustering, number of duplicated machines, and capital investments

					Efficiency of clustering			
CF	Number of EP	Number of BM	MU (%)	Number of DM	η (%)	Γ (%)	GCI (%)	Capital investments
1	7	5	87.00	0	81.78	62.50	64.52	1+2+3+4+5+6+7+8+9+10
2	5	5	91.66	0	86.45	66.70	70.97	1+2+3+4+5+6+7+8+9+10
3	5	5	92.30	1	88.20	72.72	77.42	1+2+3+4+5+6+7+8+9(2)+10
4	5	4	89.25	2	88.69	76.47	83.87	1+2+3+4+5+6+7(2)+8+9(2)+10
5	4	4	86.66	3	87.03	74.29	83.87	1+2(2)+3+4+5+6+7(2)+8+9(2)+10
6	0	0	73.80	8	86.90	73.80	100.00	1+2(3)+3(2)+4+5+6+7(2)+8(3)+9(3)+10

CF cell formation level, *EP* exceptional parts, *BM* bottleneck machines, *MU* machine utilization, *DM* Duplicate machines

grouping efficiency and grouping capability index (GCI) are 86.90% and 100.00%, respectively. From the third level to the sixth level, machine utilization decreased as the number of duplicate machine increased because the same total load was divided over a larger number of machines. Although machine utilization may be preferred in selecting the initial formation, it is likely that the cell formation with the lower machine utilization will lead to increases in production volume and demand for parts. Cell formation Levels 3 and 4 give good results in terms of machine utilization, group efficiency, and number of duplicated machines (capital investment), but the number of exceptional parts is five. These results show that there are process and routing flexibilities. Selecting one of the cell formation levels is not an easy task and depends on the management philosophy. Table 12.7 gives more analyses about manufacturing cell performance measures for Level 6 (evaluation).

The design process is terminated after finding a solution which satisfies the objectives (machine utilization, efficiency of clustering, number of exceptional parts and bottleneck machines, and machine investment) and constraints (cell size, number of cells, and number of machine types).

12.6 Conclusions

Research in the design of CMS still needs more extensive study in the areas of production and flexibility issues and cell formation techniques. There are few publications that address manufacturing flexibility and real-life production factors in cell formation when designing CMS. The need for production and flexibility

Table 12.7 Manufacturing cell performance measures for Level 6 (evaluation)

Performance measure	Value
Machine utilization MU	73.80%
Grouping efficiency η	86.90%
Grouping efficacy Γ	
Grouping capability (GCI)	100.00%
Number of part families	3
Number of machine cells	3
Number of exceptional parts	0
Number of bottleneck machines	0
Minimum cost of intercell movements	0
Minimum cost of intracell movements	NA
Cost of fixed cost	[1+2(3)+3(2)+4+5+6+7(2)+8(3)+9(3)+10]
Upper limit of cell size	7 Machines, 3 Parts
Lower limit of cell size	5 Machines, 2 Parts
Maximum number of cells	3
Maximum number of each machine type	3
Minimum number of each machine type	1

factors in designing CMS is forcing traditional manufacturing systems to be agile manufacturing systems and to cope with a changing environment. In this work, production and flexibility factors were incorporated as factors to minimize the intercellular moves and maximize the machine utilization as design objectives were restricted by several constraints to enhance the quality of the solution.

This research suggested a new heuristic cell formation approach which consisted of five main phases. In the first phase, clustering machines into machine cells was suggested through several steps based on the new comprehensive similarity coefficient between machines. The new similarity coefficient between machines was created by considering alternative routings (process plans), processing times, machine capacity (reliability), machine capability (flexibility), production volume rate, and part demand. The second phase was used to group parts into part families following many sequential steps based also on the new similarity coefficient between parts with corresponding part process plans. A new similarity coefficient between parts was also created by considering alternative routings, production volume rate, part demand, and processing times for each part.

The initial formation of the manufacturing cells was presented in the third phase after assigning part families to machine cells. In the fourth phase, performance evaluation of initial cell formation according to machine utilization and efficiency clustering was tested. A revised cell design was introduced in the fifth phase through a new strategy to eliminate exceptional parts and bottleneck machines.

References

1. Abdelmola, A.I. (2000), Modeling of Cellular Manufacturing Systems with Productivity Consideration: A Simulated Annealing Algorithm, Ph.D. Dissertation. Windsor, Canada: University of Windsor.

2. Abdelmola, A.I., Taboun, S.M., Merchawi, S. (1998), Productivity optimization of cellular manufacturing systems, Computers and Industrial Engineering, 35, 403–406.
3. Akturk, M.S., Wilson, G.R. (1998), A hierarchical model for the cell loading problem of cellular manufacturing systems, International Journal of Production Research, 36, 2005–2023.
4. Albino, V., Garavelli, A.C. (1997), Performance evaluation of a cellular manufacturing system subject to demand variability, Proceedings of the 14th International Conference on Research, 1096–1099, Osaka, Japan.
5. Albino, V., Garavelli, A.C. (1998), Some effects of flexibility and dependability on cellular manufacturing system performance, Computers and Industrial Engineering, 35, 491–494.
6. Albino, V., Garavelli, A.C. (1999), Limited flexibility in cellular manufacturing systems: A simulation study, International Journal of Production Economics, 60–61, 447–455.
7. Askin, R.G., Selim, H.M., Vakharia, A.J. (1997), A methodology for designing flexible cellular manufacturing systems, IIE Transactions, 29, 599–610.
8. Boctor, F.F. (1991), A linear formulation of the machine part cell formation problem, International Journal of Production Research, 29, 343–356.
9. Boctor, F.F. (1996), The minimum-cost, machine-part cell formation problem, International Journal of Production Research, 34, 1045–1063.
10. Brill, P.H., Mandelbaum, M. (1989), On measures of flexibility in manufacturing systems, International Journal of Production Research, 27, 747–756.
11. Carrie, A.S. (1973), Numerical taxonomy applied to group technology and plant layout, International Journal of Production Research, 11, 399–416.
12. Chan, H.M., Milner, D.A. (1982), Direct clustering algorithm for group formation in cellular manufacturing, Journal of Manufacturing Systems, 1, 65–74.
13. Chan, F.T.S. (2001), The effects of routing flexibility on a flexible manufacturing system, International Journal of Integrated Manufacturing, 14, 431–445.
14. Chandrasekharan, M.P., Rajagopalan, R. (1986), MODROC: An extension of rank order clustering for group technology, International Journal of Production Research, 24, 1221–1233.
15. Chandrasekharan, M.P., Rajagopalan, R. (1987), ZODIAC—An algorithm for concurrent formation of part families and machine cells, International Journal of Production Research, 25, 835–850.
16. Chandrasekharan, M.P., Rajagopalan, R. (1989), Groupability: An analysis of the properties of binary data matrices for group technology, International Journal of Production Research, 27, 1035–1052.
17. Chen, I.J., Chung, C.-H. (1996), An examination of flexibility measurements and performance of flexible manufacturing systems, International Journal of Production Research, 34, 379–394.
18. Chen, M. (1998), A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment, Annals of Operations Research, 77, 109–128.
19. Choi, S.-H., Kim, J.-S. (1998), A study on the measurement of comprehensive flexibility in manufacturing systems, Computers and Industrial Engineering, 34, 103–118.
20. Choobinch, F. (1988), A framework for the design of cellular manufacturing systems, International Journal of Production Research, 26, 1161–1172.
21. Chu, C.-H. (1989), Cluster analysis in manufacturing cellular formation. OMEGA, International Journal of Management Science, 17, 289–295.
22. Chu, C.-H., Lam, F.W., Lee, C.-P. (1999), Considerations for using cellular manufacturing, Journal of Materials Processing Technology, 96, 182–187.
23. Chu, C.-H., Pan, P. (1988), The Use of Clustering Techniques in Manufacturing Cellular Formation, International Industrial Engineering Conference Proceedings, 495–500, Toronto, Ontario, Canada.
24. Crama, Y., Osten, M. (1996), Models for machine-part grouping in cellular manufacturing, International Journal of Production Research, 34, 1693–1713.
25. Dahel, N.E., Smith, S.B. (1993), Designing flexibility into cellular manufacturing systems, International Journal of Production Research, 31, 933–945.
26. Drolet, J., Abdulnour, G., Rheault, M. (1996a), The cellular manufacturing evolution, Computers and Industrial Engineering, 31, 139–142.

27. Drolet, J., Rheault, M., Abdulnour, G. (1996b), Dynamic cellular manufacturing system, Computers and Industrial Engineering, 31, 143–146.
28. Dutta, S.P., Lashkari, R.S., Nadoli, G., Ravi, T. (1986), A heuristic procedure for determining manufacturing families from design-based grouping for flexible manufacturing systems, Computers and Industrial Engineering, 10, 193–201.
29. Gunasingh, K.R., Lashkari, R.S. (1989a), Machine grouping problem in cellular manufacturing systems—An integer programming approach, International Journal of Production Research, 27, 1465–1473.
30. Gunasingh, K.R., Lashkari, R.S. (1989b), The cell formation problem in cellular manufacturing systems—A sequential modeling approach, Computers and Industrial Engineering, 16, 469–476.
31. Gunasingh, K.R., Lashkari, R.S. (1991), Simultaneous grouping of parts and machines in cellular manufacturing systems—An integer programming approach, Computers and Industrial Engineering, 20, 111–117.
32. Gupta, D. (1993), On measurement and evaluation of measurement flexibility, International Journal of Production Research, 31, 2947–2958.
33. Gupta, D., Buzacott, J.A. (1989), A framework for understanding flexibility of manufacturing systems, Journal of Manufacturing Systems, 8, 89–97.
34. Gupta, T. (1991), Clustering algorithms for the design of a cellular manufacturing system—An analysis of their performance, Computers and Industrial Engineering, 20, 461–468.
35. Gupta, T. (1993), Design of manufacturing cells for flexible environment considering alternative routing, International Journal of Production Research, 31, 1259–1273.
36. Gupta, T., Seifoddini, H. (1990), Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system, International Journal of Production Research, 28, 1247–1269.
37. Han, C., Ham, I. (1986), Multiobjective cluster analysis for part family formations, Journal of Manufacturing Systems, 5, 223–230.
38. Heragu, S.S. (1994), Group technology and cellular manufacturing, IEEE Transactions on Systems, Man, and Cybernetics, 24, 203–215.
39. Ho, Y.-C., Moodie, C.L. (1996), Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities, International Journal of Production Research, 34, 2901–2923.
40. Hyer, N.L. (1984), The potential of group technology for U.S. manufacturing, Journal of Operations Management, 4, 183–202.
41. Hyer, N.L., Wemmerlov, U. (1989), Group technology in the US manufacturing industry: A survey of current practices, International Journal of Production Research, 27, 1287–1304.
42. Islam, K.M.S., Sarker, B.R. (2000), A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems, International Journal of Production Research, 38, 699–720.
43. Jeon, G., Broering, M., Leep, H.R., Parsaei, H.R., Wong, J.P. (1998a), Part family formation based on alternative routes during machine failure, Computers and Industrial Engineering, 35, 73–76.
44. Jeon, G., Leep, H.R., Parsaei, H.R. (1998b), A cellular manufacturing system based on new similarity coefficient which considers alternative routes during machine failure, Computers and Industrial Engineering, 34, 21–36.
45. Jeon, G., Leep, H.R., Parsaei, H.R., Wong, J.P. (1999), Forming part families based on alternative routes during machine failure: GA approach, International Conference of Institute of Industrial Engineers, Phoenix, AZ.
46. Joines, J.A., Culbreth, C.T., King, R.E. (1996), Manufacturing cell design: An integer programming model employing genetic algorithms, IIE Transactions, 28, 69–85.
47. Joines, J.A., King, R.E., Culbreth, C.T. (1995), A comprehensive review of production oriented manufacturing cell formation techniques, International Journal of Flexible Automation and Integrated Manufacturing, 3, 225–264.
48. Kamrani, A.K., Parsaei, H.R., Chaudhry, M.A. (1993), A survey of design methods for manufacturing cells, Computers and Industrial Engineering, 25, 487–490.

49. Kannan, V.R. (1998), Analyzing the trade-off between efficiency and flexibility in cellular manufacturing systems, Production Planning and Control, 9, 572–579.
50. King, J.R., Nakomchai, V. (1982), Machine-component group formation in group technology: Review and extension, International Journal of Production Research, 20, 117–133.
51. Kumar, V. (1987), Entopic measures of manufacturing flexibility, International Journal of Production Research, 25, 957–966.
52. Kumar, C.S., Chandrasekharan, M.P. (1990), Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, International Journal of Production Research, 28, 233–243.
53. Kusiak, A. (1987), The generalized group technology concept, International Journal of Production Research, 25, 561–569.
54. Lau, R.S.M. (1999), Critical factors for achieving manufacturing flexibility, International Journal of Operations and Production Management, 19, 328–431.
55. Lee, M.K., Luong, H.S., Abhary, K. (1997), A genetic algorithm based cell design considering alternative routing, Computer Integrated Manufacturing Systems, 10, 93–107.
56. Lozano, S., Guerrero, F., Eguia, I., Onieva, L. (1999), Cell design and loading in the presence of alternative routing, International Journal of Production Research, 37, 3289–3304.
57. Luong, L.H.S., Kazerooni, M., Abhary, K. (2001), Genetic algorithms in manufacturing system design, computational intelligence, In Manufacturing Handbook, Edited by J. Wang et al., Boca Raton, FL:CRC Press LLC.
58. Malakooti, B., Yang, Z. (2002), Multiple criteria approach and generation of efficient alternatives for machine-part family formation in group technology, IIE Transactions, 34, 837–846.
59. Mahesh, B., Srinivasan, G. (2002), Incremental cell formation considering alternative machine, International Journal of Production Research, 40, 3291–3310.
60. Marsh, R.F., Shafer, S.M., Meredith, J.R. (1999), A comparison of cellular manufacturing research presumptions with practice, International Journal of Production Research, 37(14), 3119–3138.
61. Masnata, A., Settiner, L. (1997), An application of fuzzy clustering to cellular manufacturing, International Journal of Production Research, 35, 1077–1094.
62. Miltenbury, M., Zhang, W. (1991), A comparative evaluation of nine well known algorithms for solving the cell formation problem in group technology, Journal of Operations Management, 10, 44–72.
63. Moon, C., Gen, M. (1999), A genetic algorithm-based approach for design of independent manufacturing cells, International Journal of Production Economics, 60–61, 421–426.
64. Moon, C., Kim, J. (1999), Genetic algorithms for maximizing the parts flow within cells in manufacturing cell design, Computers and Industrial Engineering, 39, 379–389.
65. Moon, C., Kim, J., Gen, M. (1999), Manufacturing Cell Design Based on Process Plans Using Genetic Algorithm, The 3rd International Conference on Engineering Design and Automation (EDA), Vancouver, 246–255, Canada.
66. Mosier, C. (1989), An experiment investigating the application of clustering procedures and similarity coefficients to the GT machine cell formation problem, International Journal of Production Research, 27, 1811–1835.
67. Mosier, C., Mahmoodi, F. (1996), Simultaneous identification of group technology machine cells and part families using a multiple solution framework, International Journal of Computer Integrated Manufacturing, 9, 402–416.
68. Mosier, C., Taube, L. (1985), The facets of group technology and their impacts on implementation—A state of the art survey, OMEGA International Journal of Management Science, 13, 381–391.
69. Moussa, S.G., Kamel, M.S. (1996), A direct method for cell formation and part-machine assignment based on operation sequences and processing time similarity, Engineering Design and Automation, 2, 141–155.
70. Mukhopadhyay, S.K., Babu, K.R., Sai, K.V.V. (2000), Modified Hamiltonian chain: A graph theoretic approach to group technology, International Journal of Production Research, 38(11), 2459–2470.

71. Mungwattana, A., Shewchuk, J. (2000), Design of cellular manufacturing systems for multiple periods using systems-dependent reconfigurations and routing flexibility, International Conference of Industrial Engineering, Cleveland, OH.
72. Nair, G.J., Narendran, T.T. (1998), CASE: A clustering algorithm for cell formation with sequence data, International Journal of Production Research, 36, 157–179.
73. Ng, S.M. (1993), Worst-case analysis of an algorithm for cellular manufacturing, European Journal of Operational Research, 69, 384–398.
74. Offodile, O.F. (1988), Application of similarity coefficient method to parts coding and classification analysis in group technology, Journal of Manufacturing Systems, 10, 442–448.
75. Offodile, O.F., Mehrez, A., Grznar, J. (1994), Cellular manufacturing: A taxonomic review framework, Journal of Manufacturing Systems, 13, 196–220.
76. Offodile, O.F., Grznar, J. (1997), Part family formation for variety reduction in flexible manufacturing systems, International Journal of Operations and Production Management, 17, 291–304.
77. Probhakaran, G., Janakiraman, T.N., Sachithanandam, M. (2002), Manufacturing data-based combined dissimilarity coefficient for machine cell formation, International Journal of Advanced Manufacturing Technology, 19, 889–897.
78. Pullen, R.D. (1976), A survey of cellular manufacturing cells, The Production Engineer, 451–454.
79. Rajamani, D., Singh, N., Aneja, Y.P. (1990), Integrated design of cellular manufacturing systems in the presence of alternative process plans, International Journal of Production Research, 28, 1541–1554.
80. Rajamani, D., Singh, N., Aneja, Y.P. (1996), Design of cellular manufacturing systems, International Journal of Production Research, 34, 1917–1928.
81. Ramabhatta, V., Nagi, R. (1998), An integrated formulation of manufacturing cell formation with capacity planning and routing, Annals of Operations Research, 77, 79–95.
82. Ravichandran, K.S., Rao, K.C.S. (2001), A new approach to fuzzy part-family formation in cellular manufacturing systems, International Journal of Advanced Manufacturing Technology, 18, 591–597.
83. Ravichandran, K.S., Rao, K.C.S., Saravanan, R. (2002), The role of fuzzy and genetic algorithms in part family formation and sequence optimization for flexible manufacturing systems, International Journal of Advanced Manufacturing Technology, 19, 879–888.
84. Reisman, A., Kumar, A., Motwani, J., Cheng, C.-H. (1997), Cellular manufacturing: A statistical review of the literature (1965–1995), Operations Research, 45, 508–520.
85. Sankaran, S. (1990), Multiple objective decision making approach to cell formation: A goal programming model, Mathematical Computation Modeling, 13, 71–82.
86. Sankaran, S., Kasilingam, R.G. (1993), On cell size and machine requirements planning in group technology systems, European Journal of Operational Research, 69, 373–383.
87. Sarker, B.R., Krishnamur, S., Kuthethur, S.G. (1994), A survey and critical review of flexibility measures in manufacturing systems, Production Planning and Control, 5, 512–523.
88. Sarker, B.R., Xu, Y. (1998), Operation sequences-based cell formation methods: A critical survey, Production Planning and Control, 9, 771–783.
89. Sarker, B.R., Xu, Y. (2000), Designing multi-product lines: Job routing in cellular manufacturing systems, IIE Transactions, 32, 219–235.
90. Sarker, B.R., Khan, M. (2001), A comparison of existing grouping efficiency measures and a new weighted grouping efficiency measure, IIE Transactions, 33, 11–27.
91. Sarker, B.R. (2001), Measures of grouping efficiency in cellular manufacturing systems: Theory and methodology, European Journal of Operational Research, 130, 588–611.
92. Seifoddini, H. (1998), Incorporation of the Production Volume in Machine Cells Formation in Group Technology Applications, Recent Developments in Production Research, Edited by A. Mital, 562–570, Elsevier Science Publishers: Amsterdam.
93. Seifoddini, H. (1989), A Note of the similarity coefficient method and the problem of improper machine assignment in group technology applications, International Journal of Production Research, 27, 1161–1165.

94. Seifoddini, H., Djassemi, M. (1991), The production data-based similarity coefficient versus Jaccard's similarity coefficient, Computers and Industrial Engineering, 21, 263–266.
95. Seifoddini, H., Djassemi, M. (1997), Determination of a flexibility range for cellular manufacturing systems under product mix variations, International Journal of Production Research, 35, 3349–3366.
96. Seifoddini, H., Hsu, C.-P. (1995), Comparative study of similarity coefficients and clustering algorithms in cellular manufacturing, Journal of Manufacturing Systems, 13(2), 119–127..
97. Seifoddini, H., Tjahana, B. (1986), Application of the similarity coefficient method in group technology, IIE Transactions, 271–277.
98. Seifoddini, H., Tjahana, B. (1999), Part-family formation for cellular manufacturing: A case study at Harnischfeger, International Journal of production Research, 37, 3263–3273.
99 Selim, H.M., Askin, R.G., Vakharia, A.J. (1998), Cell formation in group technology: Review, evaluation, and directions for future research, Computers and Industrial Engineering, 34, 3–20..
100. Sethi, A.K., Sethi, S.P. (1990), Flexibility in manufacturing: A survey, The International Journal of Flexible Manufacturing Systems, 2, 289–328.
101. Shafer, S.M., Rogers, D.F. (1991), A goal programming approach to cell formation problem, Journal of Operations Management, 10, 28–43.
102. Shafer, S.M., Rogers, D.F. (1993), Similarity and distance measures for cellular manufacturing. Part II: An extension and comparison, International Journal of Production Research, 31, 1315–1326.
103. Shambu, G. (1996), Performance evaluation of cellular manufacturing systems: A taxonomy and review of research, International Journal of Operations and Production Management, 16, 81–103..
104. Shewchuk, J.P. (1990), A Set of generic flexibility measures for manufacturing applications, International Journal of Production Research, 37, 3017–3042.
105. Shewchuk, J.P., Moodie, C.L. (2000), Flexibility and manufacturing system design: An experimental investigation, International Journal of Production Research, 38, 1801–1822..
106. Singh, N. (1993), Design of cellular manufacturing systems: An invited review, European Journal of Operational Research, 69, 284–291.
107. Singh, N., Aneja, Y.P., Rana, S.P. (1992), A bicriterion framework for operations assignment and routing flexibility analysis in cellular manufacturing systems, European Journal of Operational Research, 60, 200–210.
108. Sofianopoulou, S. (1999), Manufacturing cells design with alternative process plans and/or replicates machines, International Journal of Production Research, 37, 707–720.
109. Solymanpour, M., Vrat, P., Shankar, R. (2002), A transiently chaotic neural network approach to the design of cellular manufacturing, International Journal of Production Research, 40, 2255–2244.
110. Suer, G.A., Cedeno, A.A. (1996), A configuration-based clustering algorithm for family formation, Computers and Industrial Engineering, 31, 147–150.
111. Suer, G.A., Ortega, M. (1998), Flexibility consideration in designing manufacturing cells: A case study, In Group Technology and Cellular Manufacturing—Methodology and Applications, Edited by A. K. Kamrani and R. Logendran, Gordon and Breach Science Publishers, Vol. 1, pp. 129–152.
112. Sundaram, R.M., Doshi, K. (1992), Formation of part families to design cells with alternative routing considerations, Computers and Industrial Engineering, 23, 59–62.
113. Suresh, N.C., Slomp, J., Kaparthi, S. (1999), Sequence-dependent clustering of parts and machines: A fuzzy ART neural network approach, International Journal of Production Research, 37, 2793–2816.
114. Taboun, S.M., Merchawi, N.S., Ulger, T. (1998), Part family and machine cell formation in multi-period planning horizons of cellular manufacturing systems, Production Planning and Control, 9, 561–571.
115. Tam, K.Y. (1990), An operation sequence based similarity coefficient for part families formations, Journal of Manufacturing Systems, 9, 55–66.
116. Toni, A.D., Tonchia, S. (1998), Manufacturing flexibility: A literature review, International Journal of Production Research, 36, 1587–1617.

117. Vakharia, A.J. (1986), Methods of cells formation in group technology: A framework for evaluation, Journal of Operations Management, 6, 257–271.
118. Vakharia, A.J., Askin, R.G., Selim, H.M. (1999), Flexibility considerations in cell design, In Handbook of Cellular Manufacturing Systems, Edited by S.A. Irani, John Wiley & Sons, Inc.
119. Vakharia, A.J., Kaku, B.K. (1993), Redesigning a cellular manufacturing system to handle long-term demand changes: A methodology and investigation, Decision Sciences, 24, 909–930.
120. Venkataramanaiah, S., Krishnaiah, K. (2002), Hybrid heuristic for design of cellular manufacturing systems, Production Planning and Control, 13, 274–283.
121. Viswanathan, S. (1996), A new approach for solving the P-median problem in group technology, International Journal of Production Research, 34, 2691–2700.
122. Wemmerlov, U., Hyer, N.L. (1986), Procedures for the part family/machine group identification problem in cellular manufacturing, Journal of Operations Management, 6, 125–147.
123. Wemmerlov, U., Hyer, N.L. (1987), Research issues in cellular manufacturing, International Journal of Production Research, 25, 413–431.
124. Wemmerlov, U., Hyer, N.L. (1989), Cellular manufacturing in the US industry: A survey of users, International Journal of Production Research, 27, 1511–1530.
125. Wemmerlov, U., Johnson, D.J. (1997), Cellular manufacturing at 46 user plants: Implementation experiences and performance improvements, International Journal of Production Research, 35, 29–49.
126. Wemmerlov, U., Johnson, D.J. (2000), Empirical findings on manufacturing cell design, International Journal of Production Research, 38, 481–507.
127. Wen, H.J., Smith, C.H., Minor, E.D. (1996), Formation and dynamic routing of part families among flexible manufacturing cells, International Journal of production Research, 34, 2229–2245.
128. Wilhelm, W.E., Chiou, C.C., Chang, D.B. (1998), Integrating design and planning considerations in cellular manufacturing, Annals of Operations Research, 77, 97–107.
129. Won, Y. (2000), New P-median approach to cell formation with alternative process plans, International Journal of Production Research, 38, 229–240.
130. Won, Y., Kim, S. (1997), Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings, Computers and Industrial Engineering, 32(1), 207–220.
131. Wu, N. (1997), A concurrent approach to cell formation and assignment of identical machines in group technology, International Journal of Production Research, 36, 2005–2023.
132. Wu, N., Salvendy, G. (1999), An efficient heuristic for design of cellular manufacturing systems with multiple identical machines, International Journal of Production Research, 37, 3519–3540.
133. Yasuda, K., Yin, Y. (2001), A dissimilarity measure for solving the cell formation problem in cellular manufacturing, Computers and Industrial Engineering, 39, 1–17.
134. Yin, Y., Yasuda, K. (2002), Manufacturing cells design in consideration of various production factors, International Journal of Production Research, 40, 885–906.
135. Yoshikawa, K., Fukuta, T., Morikawa, K., Takahashi, K., Nakamura, N. (1997), A Neural Network Approach to the Cell Formation Problem, Proceedings the 14th International Conference on Production Research, 1100–1104, Osaka, Japan.
136. Zhou, M., Askin, R.G. (1998), Formation of general GT cells: An operation-based approach, Computers and Industrial Engineering, 34, 147–157.
137. Zolfaghari, S., Liang, M. (1998), Machine cell/part family formation considering processing times and machine capacities: A simulated annealing approach, Computers and Industrial Engineering, 34, 813–823.

Index

P

Q

R

S