SPRINGER OPTIMIZATION
AND ITS APPLICATIONS 41

Fatos Xhafa · Leonard Barolli
Petraq J. Papajorgji *Editors*

Complex Intelligent Systems and Their Applications

Springer

COMPLEX INTELLIGENT SYSTEMS AND THEIR APPLICATIONS

Springer Optimization and Its Applications

VOLUME 41

Aims and Scope
Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository works that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.

For other titles in this series, go to www.springer.com/series/7393

COMPLEX INTELLIGENT SYSTEMS AND THEIR APPLICATIONS

Edited By

FATOS XHAFA
Birkbeck, University of London
London, UK

LEONARD BAROLLI
Fukuoka Institute of Technology
Fukuoka, Japan

PETRAQ J. PAPAJORGJI
University of Florida
Florida, USA

Springer

Editors
Fatos Xhafa
Technical University of Catalonia
North Campus, Ed. Omega
Department of Languages &
Informatics Systems
C/Jordi Girona 1-3
08034 Barcelona
Spain
fatos@lsi.upc.edu

Leonard Barolli
Fukuoka Institute of Technology
Fac. Information Engineering
Dept. Information &
Communication Engineering
Wajiro-higashi 3-30-1
811-0214 Fukuoka
Higashi-ku
Japan
barolli@fit.ac.jp

Petraq J. Papajorgji
University of Florida
Department of Industrial &
Systems Engineering
303 Weil Hall
Gainesville, Florida 32611
USA
petraq@ise.ufl.edu

ISSN 1931-6828
ISBN 978-1-4419-1635-8 e-ISBN 978-1-4419-1636-5
DOI 10.1007/978-1-4419-1636-5
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010930775

Preface

Nowadays IT enterprises, networking, and business processes are becoming extremely demanding due to the ever-increasing complexity of systems and real-life applications. Complex intelligent systems are calling for advanced decision support systems to deal with the huge amounts of information, manipulation of complex data as well as efficiency, scalability, and security issues to support modern businesses in an autonomous, intelligent and adaptive manner.

The book *Complex Intelligent Systems and Their Applications* brings a comprehensive view of the most recent advances in complex intelligent systems and their application to the resolution of real-life problems from networking, finance, engineering, production processes, IT enterprises, and business security. The selected chapters cover a broad spectrum of issues and applications in the field of complex intelligent systems and state-of-the-art results for theoretic and practical approaches in such systems.

Among the many features of *Complex Intelligent Systems* highlighted in the book, we could distinguish the following ones by chapter:

In Chap. 1, *Moser et al.* present an approach for integrating complex information systems in the ATM domain. The large-scale and the critical issues in integration of various complex information systems in the ATM domain are real challenges tackled in the chapter. The approach presents software engineering and intelligent solutions to the integration of complex information systems in the ATM domain. An industry case is used to evaluate the approach and its comparison to traditional system integration approaches in the ATM domain.

Chapter 2 by *Veres et al.* addresses the use of semantic technologies in alignment of IT with business strategy from a requirements engineering perspective. The proposed approach is shown to be very useful in IT business. Data models and semantics are explored to achieve the goals of the proposed approach by extending BSCP (Business Strategy, Context, and Process) framework. Seven–Eleven Japan is used as a case study to validate in practice the approach.

Goebel et al. in Chap. 3 use RFID-based inter-organizational system architecture for decision support in modern business environments such as supply chain event management. By using standardized formats for event and context data, the

approach supports the interoperability of information systems in different organizations and facilitates the integration of event-based applications into enterprise architectures. Both pull- and push-based architectures are analyzed regarding efficiency and reliability.

In Chap. 4, *Hussain and Dillon* report a decision-making approach for demand-driven production processes. With the ever increasing complexity of the production processes and the demanding quality of services of costumers, the enterprises need advanced decision support systems. The proposed decision support system is aimed to hedge with third party producers to assist manufacturers in the cost-benefit analysis.

Chapter 5 by *Tashi and Ghernaouti-Hélie* proposes a security assurance model for information security in organizations. As information security is becoming very complex and critical, models for assessing assurance of security in IT enterprises is becoming imperative. In this chapter the authors bring a framework and an in-depth analysis of assurance models. Also, issues of efficiency and efficacy of the assessing the assurance are tackled.

Jakoubi et al. in Chap. 6 deal with issues arising in risk-aware business process management aiming at establishing the link between business and security. The authors present a survey of existing approaches in the literature tackling the challenge of integrating economic, risk, and security aspects. Then, a methodology enabling the risk-aware modeling and simulation of business processes is presented.

In Chap. 7, *Pournaras et al.* present AETOS (Adaptive Epidemic Tree Overlay Service), a self-organization approach for maintaining the hierarchical structures in large-scale distributed systems. The approach is shown useful in many complex applications arising in energy optimization, Internet-based multicast applications, etc. The experimental study reveals the complexity of the approach and highlights the findings, namely, ATEOS provides high connectivity in tree overlays optimized according to application requirements.

Chapter 8 by *Kitajima et al.* proposes an intelligent technique for efficiently filtering data in broadcasting systems based on the biological metaphor of attractor selection from living organisms. The approach is shown useful in many complex large-scale applications with particular focus on complex applications from networking domain. The feasibility of the proposed approach is validated by experimental study and simulations.

In Chap. 9, *Gorawski and Chrószcz* introduce a new query system for temporal data analysis. With the increasing complexity of applications and the large amounts of data to store and process, advanced query systems are a must to efficiently cope with the various challenges raised in temporal data analysis. The authors present StreamAPAS system and its declarative query language that enables users to define temporal data analysis.

Chapter 10 by *Pllana et al.* deals with agent-supported programming of multicore computing systems. The authors argue that an intelligent program development environment that proactively supports the user helps a mainstream programmer to overcome the difficulties of programming multicore computing systems. Then, a programming environment is proposed using intelligent software agents. An ex-

ample to illustrate how the best practices from HPC combined with agent-based program development can obtain efficient solutions is also given.

In Chap. 11, *Gentile and Vitabile* bring the state-of-the-art approaches in Human Computer Interaction (HCI). HCI is gaining new *momentum* due to the increasing use of a large variety of computational devices. The authors present a comprehensive view of HCI approaches and have exemplified the presentation by using agents for HCI approaches. The applicability of the approach is shown for context-aware complex distributed applications from eBusiness, Cultural Heritage, etc. for providing services and contents to costumers.

The last chapter by *Doncescu et al.* introduces new operators for advanced knowledge-based systems. Clustering has become central not only to data mining but more broadly to knowledge-based systems. The authors present novel reinforced operators that allow for using different sources of information. The approach is shown useful for advanced decision making in complex intelligent systems.

All in all, the chapters collected in this book provide new insights and approaches on the analysis and the development of *Complex Intelligent Systems* aiming to greatly support modern businesses in an autonomous, intelligent, and adaptable manner. Researchers, academics, developers, practitioners, and students will find in this book the latest trends in these research and development topics.

We hope the readers of this book will share our joy and find it a valuable resource in their research, development, and academic activities.

December 2009 The editors

Acknowledgements The editors of the book would like to thank the authors of the chapters for their contributions. We are very grateful to the referees for their feedback and constructive critique, which greatly helped improving the manuscripts. We would like to gratefully acknowledge the support and encouragement received from Prof. Panos Pardalos, the editor in chief of Springer series *Springer Optimization and Its Applications*, and Ms Elizabeth Loew (Springer, USA) for her assistance during the editorial work.

Fatos Xhafa's work is done at Birkbeck, University of London (on leave from Technical University of Catalonia, Barcelona, Spain). His research is supported by a grant from the General Secretariat of Universities of the Ministry of Education, Spain.

Contents

List of Contributors

Siegfried Benkner Department of Scientific Computing, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria, sigi@par.univie.ac.at

Stefan Biffl Complex Systems Design & Engineering Lab, Vienna University of Technology, Favoritenstr 9/188, 1040 Vienna, Austria, stefan.biffl@tuwien.ac.at

Steven Bleistein Enterprise Analysts Pty Ltd, Sydney, Australia, steve@enterpriseanalysts.com

Frances M.T. Brazier Department of Multi-actor Systems, Section Systems Engineering, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands, f.m.brazier@tudelft.nl

Aleksander Chrószcz Institute of Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland, Aleksander.Chroszcz@polsl.pl

Karl Cox University of Brighton, Brighton, UK, k.cox@brighton.ac.uk

Tharam Dillon Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia, Tharam.Dillon@cbs.curtin.edu.au

Andrei Doncescu LAAS-CNRS, University of Toulouse 7, avenue du Colonel Roche, 31077 Toulouse, France, adoncesc@laas.fr

Sergei Evdokimov Humboldt-Universität zu Berlin, Spandauer Straße 1, 10178 Berlin, Germany, evdokimov@wiwi.hu-berlin.de

Oliver Günther Humboldt-Universität zu Berlin, Spandauer Straße 1, 10178 Berlin, Germany, guenther@wiwi.hu-berlin.de

Antonio Gentile Dipartimento di Ingegneria Informatica, University of Palermo, Viale delle Scienze, Ed. 6, 90128 Palermo, Italy, gentile@unipa.it

Solange Ghernaouti-Hélie Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland, sgh@unil.ch

Christoph Goebel International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA 94703, USA, goebel@icsi.berkeley.edu

Sigrun Goluch Secure Business Austria, 1040 Vienna, Austria, sgoluch@sba-research.org

Marcin Gorawski Institute of Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland, Marcin.Gorawski@polsl.pl

Takahiro Hara Dept. of Multimedia Eng., Grad. School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan, hara@ist.osaka-u.ac.jp

Omar Hussain Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia, o.hussain@cbs.curtin.edu.au

Stefan Jakoubi Secure Business Austria, 1040 Vienna, Austria, sjakoubi@sba-research.org

Nabil Kabbaj LAAS-CNRS, University of Toulouse 7, avenue du Colonel Roche, 31077 Toulouse, France, nkabbaj@laas.fr

Shinya Kitajima Dept. of Multimedia Eng., Grad. School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan, lastis@infoseek.jp

Gerhard Kitzler Secure Business Austria, 1040 Vienna, Austria, gkitzler@sba-research.org

Eduard Mehofer Department of Scientific Computing, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria, mehofer@par.univie.ac.at

Alexander Mikula Frequentis AG, Innovationsstr. 1, 1100 Vienna, Austria, alexander.mikula@frequentis.com

Richard Mordinyi Complex Systems Design & Engineering Lab, Vienna University of Technology, Favoritenstr 9/188, 1040 Vienna, Austria, richard.mordinyi@tuwien.ac.at

Thomas Moser Complex Systems Design & Engineering Lab, Vienna University of Technology, Favoritenstr 9/188, 1040 Vienna, Austria, thomas.moser@tuwien.ac.at

Lasse Natvig Department of Computer and Information Science, NTNU, Sem Saelands vei 9, 7491 Trondheim, Norway, Lasse.Natvig@idi.ntnu.no

Shojiro Nishio Dept. of Multimedia Eng., Grad. School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan, nishio@ist.osaka-u.ac.jp

Sabri Pllana Department of Scientific Computing, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria, pllana@par.univie.ac.at

Evangelos Pournaras Department of Multi-actor Systems, Section Systems Engineering, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands, e.pournaras@tudelft.nl

Sebastien Regis Grimaag-Guadeloupe, Campus de Fouillole, B.P. 592, 97157 Pointe Pitre Cedex, France, sregis@univ-ag.fr

Jennifer Sampson Statoil, Bergen, Norway, jensam@statoil.com

Igli Tashi Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland, igli.tashi@unil.ch

Tsutomu Terada Dept. of Electrical and Electronics Eng., Grad. School of Science and Technology, Kobe University, 1-1 Rokkodai, Nada, Kobe 657-8501, Japan, tsutomu@eedept.kobe-u.ac.jp

Simon Tjoa St. Poelten University of Applied Sciences, 3100 St. Poelten, Austria, simon.tjoa@fhstp.ac.at

Christoph Tribowski Humboldt-Universität zu Berlin, Spandauer Straße 1, 10178 Berlin, Germany, christoph.tribowski@wiwi.hu-berlin.de

Csaba Veres University of Bergen, Bergen, Norway, Csaba.Veres@infomedia.uib.no

June Verner University of New South Wales, Sydney, Australia, june.verner@gmail.com

Salvatore Vitabile Dipartimento di Biopatologia e Biotecnologie Mediche e Forensi, University of Palermo, Via del Vespro, 90127 Palermo, Italy, vitabile@unipa.it

Martijn Warnier Department of Multi-actor Systems, Section Systems Engineering, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands, m.e.warnier@tudelft.nl

Fatos Xhafa Department of Computer Science and Information Systems, Birkbeck, University of London, London, UK, fatos@dcs.bbk.ac.uk

Chapter 1
Efficient Integration of Complex Information Systems in the ATM Domain with Explicit Expert Knowledge Models

Thomas Moser, Richard Mordinyi, Alexander Mikula, and Stefan Biffl

Summary The capability to provide a platform for flexible business services in the Air Traffic Management (ATM) domain is both a major success factor for the ATM industry and a challenge to integrate a large number of complex and heterogeneous information systems. Most of the system knowledge needed for integration is not available explicitly in machine-understandable form, resulting in time-consuming and error-prone human integration tasks. In this chapter we introduce and evaluate a knowledge-based approach, "Semantically Enabled Externalization of Knowledge" for the ATM domain (SEEK-ATM), which (a) explicitly models expert knowledge on specific heterogeneous systems and integration requirements and (b) allows mapping of the specific knowledge to the general ATM problem domain knowledge for semantic integration. The domain-specific modeling enables (a) to verify the integration knowledge base as requirements specification for later design of technical systems integration and (b) to provide an application program interface (API) to the problem space knowledge to facilitate tool support for efficient and effective systems integration. Based on an industry case study, we evaluate effects of the proposed SEEK-ATM approach in comparison to traditional system integration approaches in the ATM domain. Major advantages of the novel approach are the efficient derivation of technical configurations and automated quality assurance of the expert knowledge models.

T. Moser (✉) · R. Mordinyi · S. Biffl
Complex Systems Design & Engineering Lab, Vienna University of Technology,
Favoritenstr 9/188, 1040 Vienna, Austria
e-mail: thomas.moser@tuwien.ac.at

R. Mordinyi
e-mail: richard.mordinyi@tuwien.ac.at

S. Biffl
e-mail: stefan.biffl@tuwien.ac.at

A. Mikula
Frequentis AG, Innovationsstr. 1, 1100 Vienna, Austria
e-mail: alexander.mikula@frequentis.com

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_1, © Springer Science+Business Media, LLC 2010

1.1 Introduction and Motivation

In the Air Traffic Management (ATM) domain complex information systems need to cooperate to provide data analysis and planning services, which consist in the core of safety-critical ATM services and also added-value services for related businesses. ATM is a relevant and dynamic business segment with changing business processes that need to be reflected in the integration of the underlying information and technical systems.

A major integration challenge is to explicitly model the knowledge embedded in systems and ATM experts to provide a machine-understandable knowledge model for integration requirements between a set of complex information systems (CIS). CIS consist of a large number of heterogeneous subsystems. Each of these subsystems may have different data types and heterogeneous system architectures. In addition, CIS typically have significant quality-of-service demands, e.g., regarding security, reliability, timing, and availability. Many of today's ATM CIS were developed independently for targeted business needs, but when the business needs changed, these systems needed to be integrated into other parts of the organization (Halevy 2005). Most of the system knowledge is still represented implicitly, either known by experts or described in human-only-readable sources, resulting in very limited tool support for systems integration. The process of establishing and/or maintaining integration solutions of business systems is traditionally a human-intensive approach of experts from the ATM and technology domains.

Making the implicit expert knowledge explicit and understandable for machines can greatly facilitate tool support for systems integrators and engineers by providing automation for technical integration steps and automatic validation of integration solution candidates. The overall process for systems integration consists of three phases (see Moser et al. 2009): (1) the elicitation and validation of systems integration requirements (problem space knowledge); (2) the description of the architecture and the modeling of the capabilities of technical solution candidates (solution space knowledge) (Mordinyi et al. 2009); and (3) the bridging of the knowledge models of problem and solution space to identify the most suitable solution candidates (Moser et al. 2009).

In this chapter we focus on the first phase of system integration to provide the foundation for the later phases. We propose a knowledge-based approach, "Semantically Enabled Externalization of Knowledge" for the ATM domain (SEEK-ATM), which (a) explicitly models specific heterogeneous system and expert knowledge on integration requirements using a three-layered ontology architecture for storing knowledge, (b) allows mapping of the specific knowledge to the general ATM problem domain knowledge for enabling semantic integration, and (c) facilitates tool support for, e.g., requirements validation by means of providing homogeneous access to heterogeneous integration knowledge. The knowledge base provides tool access to knowledge models based on a common problem domain model, allowing queries or validation of heterogeneous knowledge sources. The output of this phase is a validated knowledge base of business requirements for integration as input to technical design steps.

We evaluate the effectiveness and efficiency of the SEEK-ATM approach in an industrial case study in the ATM domain. Based on two integration scenarios, we determine key performance indicators, like integration effort, integration duration, quality assurance efficiency, model complexity, and level of automation support in order to compare the SEEK-ATM approach with traditional system integration approaches in the ATM domain.

The remainder of this chapter is structured as the following: Sect. 1.2 motivates research issues and pictures the use case, Sect. 1.3 summarizes related work, Sects. 1.4 and 1.5 describe the SEEK-ATM approach, Sect. 1.6 presents evaluation results. Finally, Sect. 1.7 concludes and gives an outlook on future research work.

1.2 Objectives and Contribution

Recent projects with industry partners from the safety-critical ATM domain raised concerns about the verification of modern technology-driven integration environments. For certification, a major goal was to improve the capability of engineers to verify an integration solution by facilitating team work and tool support.

The data-driven SEEK approach (Moser et al. 2009) has been developed in order to explicitly model the semantics of the problem space, the solution space, and provide a process to bridge problem and solution spaces. The SEEK approach, described in Moser et al. (2009) in more details, consists of 6 process steps: (1) legacy system description, (2) domain knowledge description, (3) model QA, (4) derivation and selection of integration partners, (5) generation of transformation instructions, and (6) configuration QA. For a typical systems integration scenario, the problem space is described as integration requirements and capabilities, the solution space consists of connectors and data transformation instructions between legacy systems, while the bridging process between both spaces is concerned with finding feasible integration solutions, e.g., with minimal integration costs.

In this chapter, we apply the original SEEK process to a use case example from the ATM domain and describe the resulting variant of the SEEK process, SEEK-ATM, with a main focus on the first three process steps, namely the modeling of integration requirements and capabilities for integration knowledge elicitation and QA, resulting in the following research issues.

RI-1. *Foundations for Tool Support for Automation of Integration Steps*. Investigate to what extent (e.g., effort saved during process execution) the explicit and machine-understandable semantic modeling of integration knowledge helps to automate time-consuming systems integration steps. Investigate the effect of the automated integration process steps regarding the quality assurance efficiency. As precondition for RI-1, we needed to ensure that (a) the knowledge is complete enough for relevant tool support, and (b) the knowledge can be accessed by tools, e.g., by means of an API.

RI-2. *More Efficient and Effective Systems Integration Process Steps*. Investigate whether the SEEK-ATM approach provides an overall more efficient and effective systems integration process regarding key performance indicators like integration

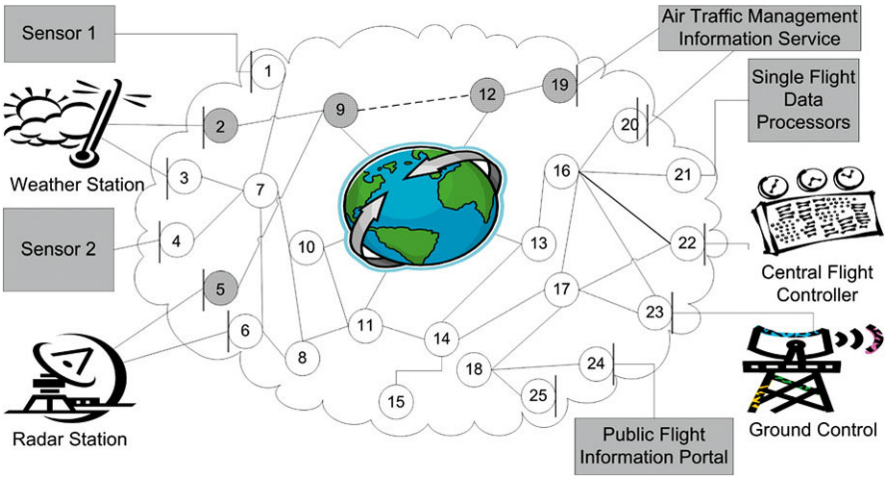


Fig. 1.1 Overview use case example: network between information providers and consumers

effort and duration, QA efficiency, model complexity, and level of automation support.

For empirical evaluation, we determine the integration effort needed for each process step to compare the steps in the new SEEK-ATM approach with traditional methods and measure the effectiveness and efficiency of the available methods and tools.

A requirement of the ATM domain is to provide timely and correct data analyses from a web of heterogeneous legacy applications. The high number of distributed legacy applications with heterogeneous interfaces to their services on the one hand and the need to dramatically improve the flexibility in order to provide new ways of systems integration in a safety-critical environment on the other hand, demanded for a innovative approach like the SEEK-ATM.

The ATM use case (Fig. 1.1) represents information that is typically extracted from participants in workshops on requirements elicitation for information systems in the aviation domain. The business system ATM Information Service (ATMIS) has to provide information services about flights to business partners via a Public Flight Information Portal (PFIP). ATMIS needs to collect and refine information from at least two other systems: the Central Flight Controller (CFC) and the Single Flight Data Processors (SFDPs). As input to integration process, each data provider, in our case CFC and SFDPs, defines the data content and format he can provide and the quality of service, e.g., the frequency of incoming data such as radar signals; each data consumer, in our case ATMIS, similarly defines his needs for data content, format, and quality of service and may additionally require conditions such as data coming from a defined geographical area and within a defined time window. Finally, the network provider describes the capacity of connectors between the data provider and consumer nodes, and the quality of service of these connectors, e.g., security levels, reliability. All systems have requirements on reliability, timeliness, safety,

service quality, failover, performance, auditability, maintainability, and flexibility. An additional requirement regarding a possible systems integration solution is the capability of agile reaction to any kind of changes due to altered business needs.

Figure 1.2 illustrates traditional approaches to systems integration in the ATM domain: There are database-style and/or UML models of the systems interfaces, which work well together in a homogeneously designed set of systems. However, in typical domains the systems often exhibit heterogeneous semantics, i.e., similar meaning can be expressed in several ways. Currently, highly skilled domain experts in the ATM problem space and the technical solution space bridge these semantics as there are so far no machine-readable models available to facilitate comprehensive tool support. However, the limited availability of these experts slows down the pace of strategically desirable integration projects.

1.3 Related Work

This section summarizes related work on semantic integration using ontologies.

1.3.1 Semantic Data Integration

Semantic Integration is defined as the solving of problems originating from the intent to share data across disparate and semantically heterogeneous data (Halevy 2005). These problems include the matching of ontologies or schemas, the detection of duplicate entries, the reconciliation of inconsistencies, and the modeling of complex relations in different sources (Noy et al. 2005). Over the last years, semantic integration became increasingly crucial to a variety of information-processing applications and has received much attention in the web, database, data-mining, and AI communities. One of the most important and most actively studied problems in semantic integration is establishing semantic correspondences (also called mappings) between vocabularies of different data sources (Doan et al. 2004).

Goh (1996) identified three main categories of semantic conflicts in the context of data integration that can appear: confounding conflicts, scaling conflicts, and naming conflicts. The use of ontologies as a solution option to semantic integration and interoperability problems has been studied over the last 10 years. Wache et al. (2001), reviewed a set of ontology-based approaches and architectures that have been proposed in the context of data integration and interoperability.

Noy (2004) identified three major dimensions of the application of ontologies for supporting semantic integration: the task of finding mappings (semi-) automatically, the declarative formal representation of these mappings, and reasoning using these mappings. There exist two major architectures for mapping discovery between ontologies. On the one hand, the vision is a general upper ontology which is agreed upon by developers of different applications. Two of the ontologies that are built specifically with the purpose of being formal top-level ontologies are the Suggested

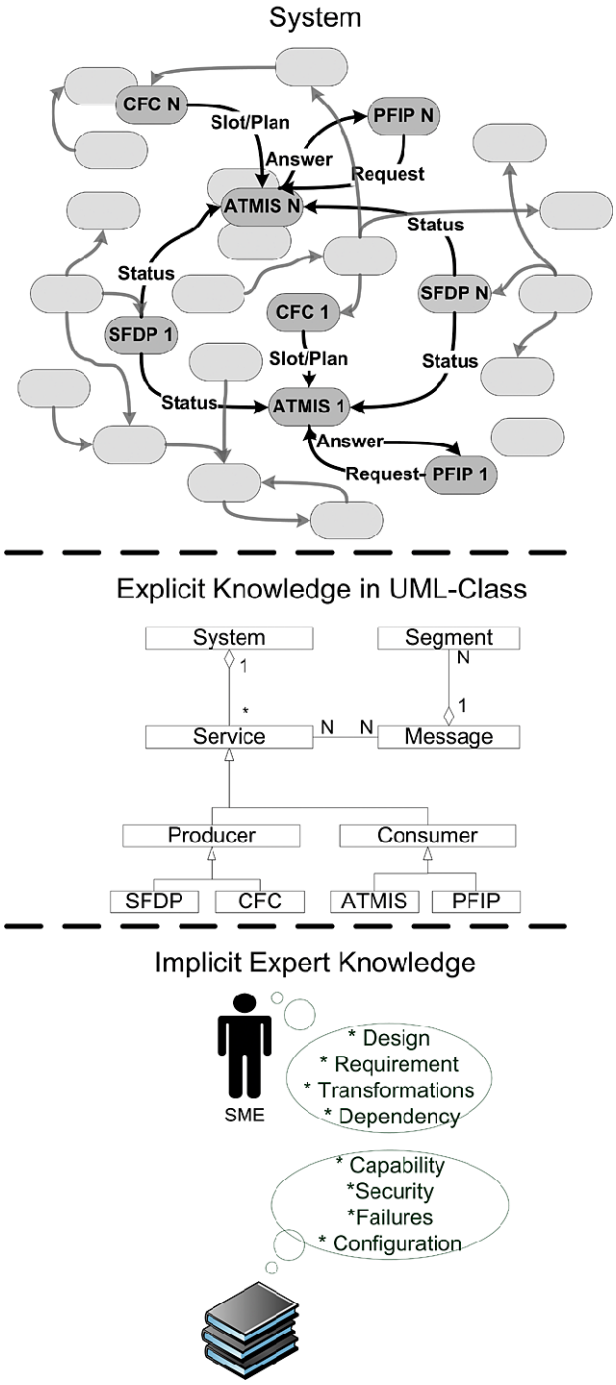


Fig. 1.2 Air traffic management systems integration—explicit and implicit expert knowledge

Upper Merged Ontology (SUMO) (Niles and Pease 2001) and DOLCE (Gangemi et al. 2003). On the other hand, there are approaches comprising heuristics-based or machine learning techniques that use various characteristics of ontologies (e.g., structure, concepts, instances) to find mappings. These approaches are similar to approaches for mapping XML schemas or other structured data (Bergamaschi et al. 1999; Cruz et al. 2004). The declarative formal representation of mappings is facilitated by the higher expressive power of ontology languages which provide the opportunity to represent mappings themselves in more expressive terms. There exists a large spectrum of how mappings are represented. Bridging axioms relate classes and properties of the two source ontologies and can be seen as translation rules referring to the concepts of source ontologies and, e.g., specifying how to express a class in one ontology by collecting information from classes in another ontology. Another mapping representation is the declarative representation of mappings as instances in an ontology. This ontology can then be used by tools to perform the needed transformations. Then a mapping between two ontologies constitutes a set of instances of classes in the mapping ontology and can be used by applications to translate data from the source ontology to the target. Naturally, defining the mappings between ontologies, either automatically, semi-automatically, or interactively, is not a goal in itself. The resulting mappings are used for various integration tasks: data transformation, query answering, or web-service composition, to name a few. Given that ontologies are often used for reasoning, it is only natural that many of these integration tasks involve reasoning over the source ontologies and the mappings.

1.3.2 Ontologies for Semantic Integration

Ontologies can support data integration processes by providing a continuous-data model (Calero et al. 2006) that helps bridging semantic gaps between systems and/or processes. Compared to traditional common data models like UML Class Diagrams or Entity Relationship Diagrams (ERDs), ontologies both (a) provide methods for integrating data models using automated transformation and (b) support the concurrent modeling of different systems (Hepp et al. 2007). There is a wealth of research reports on the extension of UML to support Ontology Engineering for the Semantic Web (Baclawski et al. 2001). For Quality Assurance (QA), ontologies can check whether a model has knowledge missing or inconsistent knowledge.

There has been ample research (Happel and Seedorf 2006) on the use of ontologies for supporting typical software engineering processes like systems integration. Ontology-Driven Architecture (ODA) is introduced, serving as a starting point for the W3C to elaborate a systematic categorization of the different approaches for using ontologies in Software Engineering. The current MDA-based (Miller and Mukerji 2001) infrastructure provides architecture for creating models and meta-models (e.g., models of the systems to be integrated), define transformations between those models (e.g., transformations between integrated systems), and managing metadata. Though the semantics of a model is structurally defined by its meta-model, the

mechanisms to describe the semantics of the domain are rather limited compared to knowledge representation languages. In addition, MDA-based languages do not have a knowledge-based foundation to enable reasoning (e.g., for supporting QA) (Baclawski et al. 2002). System integration can benefit from the integration with ontology languages such as RDF and OWL (Gasevic et al. 2005, 2006) in various ways, e.g., by reducing language ambiguity, enabling validation, and automated consistency checking.

Uschold et al. (2004) identified four main categories of ontology application to provide a shared and common understanding of a domain that can be communicated between people and application systems (Fensel 2003): Given the vast number of noninteroperable tools and formats, a given company or organization can benefit greatly by developing their own neutral ontology for authoring, and then developing translators from this ontology to the terminology required by the various target systems. To ensure no loss in translation, the neutral ontology must include only those features that are supported in all of the target systems. The trade-off here is loss of functionality of some of the tools, since certain special features may not be usable. While it is safe to assume there will not be global ontologies and formats agreed by one and all, it is nevertheless possible to create an ontology to be used as a neutral interchange format for translating among various formats. This avoids the need to create and maintain $O(N^2)$ translators for N systems, and it makes it easier for new systems and formats to be introduced into an existing environment. In practical terms, this can result in dramatic savings in maintenance costs—it has been estimated that 95% of the costs of enterprise integration projects is maintenance (Pollock 2002).

There is a growing interest in the idea of "Ontology-Driven Software Engineering" in which an ontology of a given domain is created and used as a basis for specification and development of some software. The benefits of ontology-based specification are best seen when there is a formal link between the ontology and the software. This is the approach of Model-Driven Architecture (MDA) (Miller and Mukerji 2001) created and promoted by the Object Modeling Group (OMG) as well as ontology software which automatically creates Java classes and Java Documents from an ontology. A large variety of applications may use the access functions of the ontology. Not only does this ensure greater interoperation, but it also offers significant cost reduction for software evolution and maintenance. A suite of software tools all based on a single core ontology are semantically integrated for free, eliminating the need to develop translators. To facilitate search, an ontology is used as a structuring device for an information repository (e.g., documents, web pages, names of experts); this supports the organization and classification of repositories of information at a higher level of abstraction than is commonly used today. Using ontologies to structure information repositories also entails the use of semantic indexing techniques, or adding semantic annotations to the documents themselves. If different repositories are indexed to different ontologies, then a semantically integrated information access system could deploy mappings between different ontologies and retrieve answers from multiple repositories.

1.4 Making Integration Knowledge Explicit

This section pictures the semantic modeling of heterogeneous knowledge using a set of ontologies as model. The ontology architecture (Moser et al. 2007) is described in detail as well as the distribution of the modeled information among the layers.

The ontologies used as input models for the derivation of the system configuration are organized using a subdivided architecture, consisting of three different types of ontologies. The ontology types building the semantic model for a specific scenario are the Abstract Integration Scenario Ontology (AISO), the Domain-specific Ontologies (DSO), and the Integration System Ontologies (ISO) (see Fig. 1.3). The DSOs extend the AISO by adding concepts describing the common domain knowledge used. In addition, the ISO uses the other two ontologies for aligning its concepts with the more general concepts defined in either the AISO or DSO.

1.4.1 Abstract Integration Scenario Ontology

The Abstract Integration Scenario Ontology (AISO) is defined in an application-domain-independent manner, allowing its use across different domains. This domain-independent definition is a powerful mechanism to provide a flexible base for information sharing scenarios, completely independent of a particular domain. The terms in the AISO are defined in an abstract way to simplify the conceivability of the use in different domains.

1.4.2 Domain-Specific Ontology

The Domain-Specific Ontology (DSO) includes the main shared knowledge between stakeholders of the particular domain (e.g., ATM domain) and hence rep-

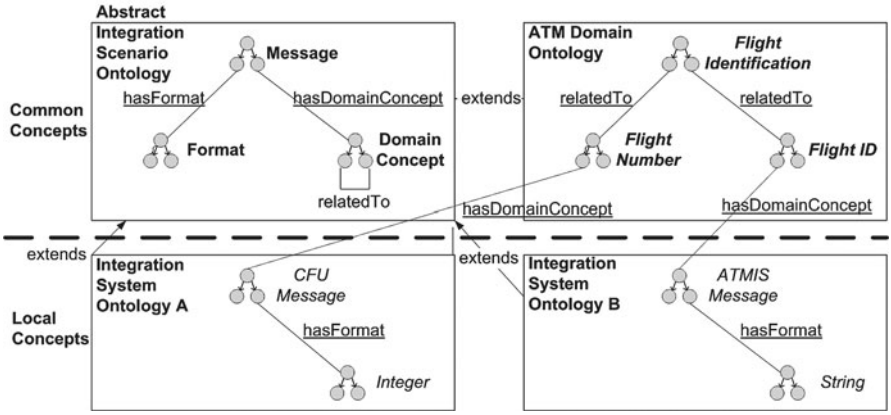


Fig. 1.3 Simplified ontology architecture example (Moser et al. 2007; Moser et al. 2009)

resents the collaborative view on the information exchanged in an integration scenario. In addition, the DSO is the place to model standardized domain-specific information. The customers map their proprietary information, which is defined in the integration system ontologies, to the standardized information in order to allow the interoperability with other participants.

This domain-specific information is used for the detection of semantically identical information provided or consumed by participating applications or organizations, independent of the format or identifiers used for the information, and therefore improves or enables the communication between these organizations. The identification of possible integration partners is simplified, and the tool-supported transformation of semantically identical information existing in different formats allows further communication between new partners.

This particular domain-specific knowledge described in the DSO can easily be updated or transferred to other SEEK-ATM approach-based integration scenarios residing in the same domain. This allows a broad spectrum of new applications in a particular domain to benefit from the described domain knowledge. Instead of modeling the domain knowledge from scratch, it is also possible to use as starting point a description of the problem domain, a so-called "world model." The advantage of this approach is the reduced effort for modeling the domain knowledge; however a tradeoff exists in the complexity of typical "world model" ontologies, resulting in a longer waiting time when searching for concrete domain knowledge.

1.4.3 Integration System Ontology

The Integration System Ontology (ISO) defines the customer-specific, proprietary view on the information exchanged in an integration scenario. This includes the view on the format of the information (as required by the legacy application) but can also describe the meaning or the use of the specific view on the existing information, since there can exist multiple views for the same information. The ISO defines the structure of the legacy applications, services, and messages, i.e., the services provided by a legacy application, the messages provided or consumed by a service, and the message segments a message consists of, by adding instances of the concepts defined in either the AISO or the DSO.

The most important part of this description is the definition of the exchanged information, i.e., the definition of the messages either provided or consumed by the legacy applications. The ISO describes the semantic context and the format of each message segment supported by the domain expert. Each message segment is mapped to exactly one particular domain concept. This defines the semantic context of the information contained in the segment and allows the detection of possible collaborations for an integration scenario. In addition, the format of the information is described, enabling automated transformation between formats.

1.5 SEEK-ATM Process Description

This section summarizes the key factors of the SEEK-ATM approach. Figure 1.4 gives a short overview of the SEEK-ATM process steps for requirements elicitation and validation in comparison with a traditional integration approach.

Traditional Integration Approach In the traditional integration approach, for each legacy information system to be integrated, the Subject Matter Expert (SME) responsible for the particular system describes the requirements and capabilities of the system using human-readable (but typically not machine-readable) language. The outcome of this process step is a set of legacy systems interface description documents. The QA step is performed mostly by humans and mainly consists of (a) a comparison of the knowledge represented in the legacy systems interface description documents with the knowledge captured implicitly by the SMEs and (b) a comparison of the accepted set of integration partners and the needed transformation instructions with the knowledge represented in the legacy systems interface description documents and again with the knowledge captured implicitly by the SMEs. In the traditional integration process, there are 2 QA steps performed mostly by humans: (a) comparison of the knowledge represented in the legacy systems interface description documents with the knowledge captured implicitly by the SMEs; (b) comparison of the accepted set of IPs and the needed transformation instructions with the knowledge represented in the legacy systems interface description documents and again with the knowledge captured implicitly by the SMEs. As key parts of this knowledge are not available in machine-readable form, tool support for QA is very limited and takes much effort from scarce human experts.

SEEK-ATM Integration Approach In the SEEK-ATM approach, for each legacy information system to be integrated, the SME responsible for the particular system describes the requirements and capabilities (R&Cs) of the system using machine-readable notation. In addition to these R&Cs, the semantic meaning of the exchanged information is externalized by mapping information to more general knowledge represented in the domain ontology. In comparison to the traditional integration process, the outcome of this process step is a set of ontologies describing the R&Cs of the legacy information system to be integrated and the mapping of the information to general domain knowledge. In addition to the description of the R&Cs of the participating systems, the domain expert (DE) describes the common knowledge of the problem domain used in the integration scenario.

This externalized domain knowledge is used by the SMEs while describing the particular legacy systems. The outcome of this process step is an ontology describing the shared domain knowledge of the problem domain used in the integration scenario. This domain ontology can be reused for a set of different integration scenarios in a domain. The QA step in the SEEK-ATM integration approach can be very well supported with tools based on ontology-based reasoning. Reasoning allows checks for consistency (e.g., whether information entered in different input

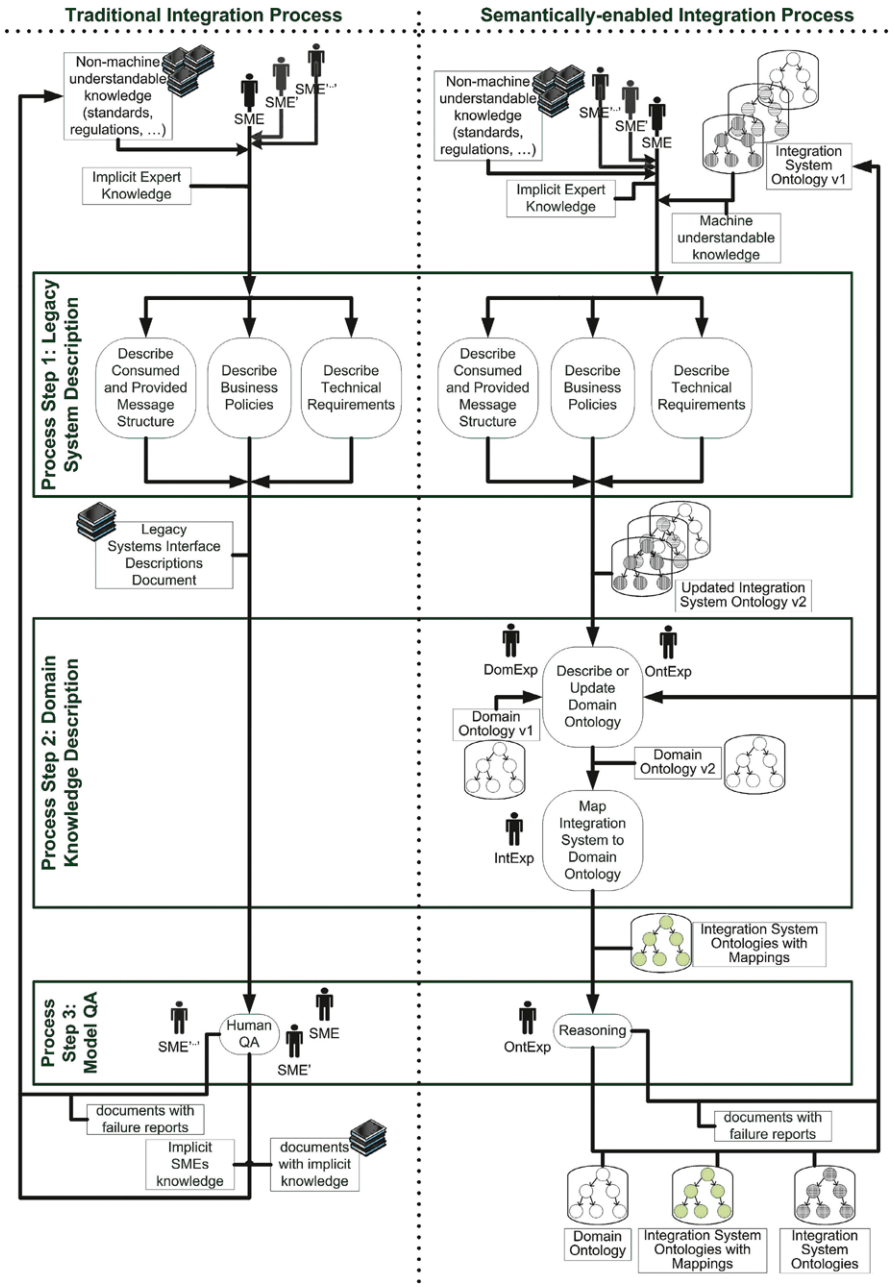


Fig. 1.4 Side-by-side comparison of the traditional and the SEEK-ATM integration process steps

masks is consistent) and completeness (e.g., whether all needed information is entered). This allows a much faster and more reliable QA compared to the traditional integration process and allows relieving scarce experts from tedious work.

To summarize the process description, for both the traditional integration process and the SEEK-ATM process, the input is the same, but the output differs.

While the output of the traditional integration process still consists of mainly implicit knowledge, the output of the SEEK-ATM process consists of explicit and machine-understandable knowledge.

1.6 Added Value from Explicit Knowledge

This section pictures usage scenarios for heterogeneous knowledge integrated using the SEEK-ATM approach. In addition, the real usage scenarios from the exemplary ATM use case are described shortly. The knowledge can be used for a set of queries like checking the consistency of the integrated data (e.g., by measuring type similarity between concepts) or checking the completeness of the mapped concepts (e.g., whether it is possible to fulfill the given requirements with the modeled knowledge).

In the use case from the ATM domain, the integrated knowledge can be used for the automated identification of integration partner candidates, the generation of transformation instructions and for the generation of system integration configurations. This allows a much faster and more reliable QA compared to the traditional integration process and relieves scarce experts from tedious work. The following paragraphs summarize these usage examples.

Automated Identification of Integration Partner Candidates For every consumer service, the set of possible provider services providing the required information is calculated. These sets of pairs of a consumer service and at least one provider service, together with the required transformation instructions, are called collaboration candidates. The Domain Expert (DomExp) and the customer SMEs choose one or, if applicable, more desired collaborations from these collaboration candidates. Then the system integration configuration for these chosen collaborations is calculated by the SEEK-ATM approach. The externalized knowledge of the SMEs, the DomExp, and the Network Administrator (NA), which is captured in the ontologies created in the previous steps, is used to automatically derive the set of possible integration partners using ontology-based reasoning, allowing an easier and less error prone identification of possible integration partners compared to the traditional integration process. The outcome of this process step is a set of possible integration partners. The Integration Expert (IntExp) is responsible for choosing the wanted integration partners from the set of possible integration partners derived in the previous step. The outcome of this process step is a set of accepted integration partners.

Generation of Transformation Instructions After these integration partners are selected, the transformation instructions for these collaborations need to be created.

This generation process is semiautomatic and supervised by the DomExp. The DomExp reviews the generated transformation instructions and has to accept it in order to be functional.

Generation of System Integration Configuration The information derived in the previous steps is used to create the final system integration configuration. The configuration is stored in an XML file containing information on all the needed instructions to run the system, such as routing tables, transformation instructions, and binding descriptions for connecting to particular legacy systems.

1.7 Evaluation

In the previous section RI-1 has been addressed. To discuss the RI-2, we started an evaluation by means of the proposed entire SEEK-ATM approach. Therefore, we derived four parameters (see Table 1.1) to compare the proposed approach with the traditional one. Table 1.1 summarizes the effort and duration needed for integration, the quality assurance efficiency, the complexity of the used models, and finally the level of automation support both approaches provide.

The evaluation is based on two scenarios within the ATM use case. The first scenario (Sc. 1) determines the results based on an integration project from the scratch. The second scenario (Sc. 2) assumes that an initial integration project has been accomplished providing a first integration solution, but due to changing business requirements, some system adaptations have to be performed, like the need to update the domain model. Scenario 1 within the ATM use case has the following characteristics: five systems (applications) with 30 integration points (services) and 100 data structures (logical entities). In case of Sc. 2, 10 integration points of three different systems have been updated resulting in two new data structures and 10 updated ones. The overall integration effort for scenario 1 using the traditional approach was 415 PDs[1] and, for scenario 2, 76 PDs. When using the SEEK-ATM approach, the overall integration effort for scenario 1 was 435 PDs, compared to 32 PDs for scenario 2.

Integration Effort The results of the evaluation show that the overall integration effort is similar for both approaches in case of small number of systems to be integrated and slightly higher for the SEEK-ATM approach in case of larger systems. The higher effort comes from the need to manage the domain model, since additional mappings between the integration system ontology and the domain model are needed. The effort to create the integration system ontology or the interface description is similar since in both approaches the conducted SMEs have to cope with the same problem of finding the right information describing the system interfaces with its semantics. The SEEK-ATM has the advantage that in case of adaptation

[1] PD: Person Day (Full Time Equivalent).

Table 1.1 Comparison of the traditional and the SEEK-ATM approaches

Evaluation parameters	Traditional approach	SEEK-ATM approach
Integration effort	System knowledge is described in human-readable documents by Subject Matter Experts (SMEs).	System knowledge is externalized in a machine-readable ontology by SMEs.
	No explicit domain knowledge used.	Domain knowledge is incrementally externalized in a machine-readable ontology by the Domain Expert (DomExp).
QA efficiency	Low	High
	Manual checks of documents and models needed (time consuming and error prone).	Automated ontology reasoning allows quickly locating inconsistent knowledge in the model.
Model complexity	High and distributed	High and centralized
Level of automation support	Low	High
	Exhaustive communication between SMEs, DomExp, and IntExp is needed to clarify dependencies and integration partners.	Automated derivation of possible integration partners by means of ontology based reasoning.
	DomExp coordinates the generation of transformation instructions with the affected SMEs.	Automated derivation of transformation instructions by means of ontology based reasoning.
	Manual checks of documents and system configuration needed (time consuming and error prone).	Automated ontology reasoning allows quickly locating invalid system configurations.

the knowledge already gathered is explicitly given and can be reused in further discussions compared to the traditional approach where this knowledge exists implicitly only. In case of reconfiguration issues the SEEK-ATM process has proven to be more efficient than the traditional approach since once the knowledge has been externalized, it can be reused with little extra effort. Furthermore, in case of the traditional approach each system expert has to be contacted for any kind of changes resulting in discussions.

In case of the SEEK-ATM approach the domain expert is needed in major changes only where the mapping of the integration system ontology to the domain ontology has to be altered as well. In case of minor changes, affecting the characteristics of the system only, the SMEs are needed. Additionally, performing changes, like structure modifications, based on documents is more difficult and time consuming than compared with ontologies where you deal with classes. Changes can be performed much faster and can be done during the discussion concerning the integration project as well. The duration of the traditional approach tends to be higher due to error-prone mainly manual process steps resulting in additional efforts to

discuss error sources and possible solutions. The proposed SEEK-ATM approach reports errors or missing information immediately due to in-time consistency and completeness checks based on ontology reasoning. In case of describing systems, parallel processing is possible in both approaches. However, the following SEEK-ATM processing steps are running mainly automated from the third processing step on, while the traditional approach is still human-driven resulting in time consuming and error-prone processing steps. Therefore, the duration depends strongly on of automation support.

QA Efficiency Since the traditional approach focuses on manual validity checks, it is therefore more time consuming and error prone. This also results in the fact that missing information is often detected in a later integration step. The quality assurance efficiency is measured by the number of failures detected in each system description weighted by the time of detection. The later the failure detected, the higher the weighting rate. The SEEK-ATM approach uses ontology-based reasoning. This allows performing consistency and completeness checks in-time automatically, resulting in a lower failure rate and in-time notification of the SME about missing/incorrect information. Additionally, since the SEEK-ATM approach is mainly automated, it allows returning to any processing state in order to, e.g., reproduce errors or revise decisions taken.

Model Complexity The model used in the traditional approach is smaller and therefore less complex compared to the model used in the SEEK-ATM approach, since a considerable part of the integration knowledge is not described explicitly. In the SEEK-ATM approach, the number of relations, i.e., the number of mappings from the integration system ontology to the domain ontology introduces a higher structural complexity. The benefit of a more complex ontology model lies in the way how later integration steps can be supported by a higher level of automation. From the SME's point of view the complexity remains the same in both approaches. For the domain expert, the SEEK-ATM approach reduces his efforts to the task of managing the structural complexities of the ontologies and to support the SMEs in mapping. In the traditional way the domain experts need to cope with the major part of the complexity, since they are responsible for ensuring the consistency and completeness as well as managing the integration of the SMEs' legacy system descriptions.

Level of Automation Support The SEEK-ATM approach supports the user while entering the data with consistency and completeness checks. Additionally, it influences the integration process in later steps by automatically deriving integration partner candidates and automatically generating transformation instructions for message exchange between the integrated systems.

Within a research project with two industry partners, the approach has been evaluated by means of several different scenarios from the ATM domain. We determine the effort for both process step variants and compare the overall outcome. The following paragraphs summarize the effort needed to perform the particular process

steps. The effort estimations are based on the expertises of the integration experts from both companies.

Step 1. *Legacy System Description*: The externalization of legacy system knowledge using ontologies needs slightly more effort than the traditional approach using only human-readable artifacts like documents because the knowledge needs to be transformed from implicit expert or system knowledge into machine-readable ontology models.

Step 2. *Domain Knowledge Description*: In the traditional integration process the domain knowledge is not made explicit but implicitly captured by domain experts and documents in a non-machine-readable way requiring no additional effort. Additionally, the integration network knowledge (i.e., the architecture and capabilities of the underlying network infrastructure) are described, which again represent an additional effort compared to the implicit knowledge of the traditional integration process. Using the *SEEK* approach, the domain and integration network knowledge has to be incrementally externalized by the domain expert and the network administrator, resulting in medium effort in the first instance. This effort is reduced due to reuse within similar integration scenarios or additional process iterations triggered by reconfiguration issues.

Step 3. *Model QA*: The traditional approach requires a lot of effort to check the consistency and completeness of the documents since it has to be done manually. The *SEEK* approach uses automated ontology-based reasoning techniques to assure consistent models leading to comparatively low model QA effort.

1.8 Conclusion and Future Work

In this chapter, we introduced and evaluated a domain-specific approach for ATM to make expert knowledge on heterogeneous systems and system integration requirements explicit to facilitate tool-support for design and QA. An important contribution of the chapter is to enable new research and application areas for semantic techniques that help control complex information system. Major results of our research evaluation of SEEK-ATM in an industrial case study were:

1. *Tool support for automation of integration steps*. The explicit and machine-understandable knowledge in SEEK-ATM helps to automate time-consuming systems integration steps like consistency and completeness checks. Furthermore, it allows automating later integration processing steps, like deriving integration partner candidates or automatically generating transformation instructions for message exchange between the integrated systems.

2. *More efficient and effective systems integration*. The evaluation showed that the integration effort needed with the SEEK-ATM approach is slightly higher in case of integration from the scratch, but comparatively a lot smaller when adaptations due to changing business need have to be performed. In addition, the advantage of centrally storing the domain ontology together with the mappings of individual system knowledge lies in the possibility of an automated QA and automation of further integration steps resulting in less integration efforts and less failures.

Further work will extend the semantic modeling of the problem space to the technical solution space and ultimately ways to bridge problem and solution spaces, as well as to include a large-scale evaluation of the SEEK-ATM approach using scenarios and integration effort measurements in real-world integration projects. Additionally, the feasibility of ontology-based reasoning for usage in QA will be evaluated.

References

K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski, and M. Aronson, "Extending UML to support ontology engineering for the semantic web," *4th International Conference on UML*, Springer, Berlin, 2001, pp. 342–360.

K. Baclawski, M.K. Kokar, P.A. Kogut, L. Hart, J. Smith, J. Letkowski, and P. Emery, "Extending the unified modeling language for ontology development," *International Journal of Software and Systems Modeling (SoSyM)*, vol. 1, no. 2, 2002, pp. 142–156.

S. Bergamaschi, S. Castano, and M. Vincini, "Semantic integration of semitructured and structured data sources," *SIGMOD Record*, vol. 28, no. 1, 1999, pp. 54–59.

C. Calero, F. Ruiz, and M. Piattini, *Ontologies for Software Engineering and Software Technology*, Springer, Berlin, 2006.

I.R. Cruz, X. Huiyong, and H. Feihong, "An ontology-based framework for XML semantic integration," *International Database Engineering and Applications Symposium (IDEAS '04)*, IEEE, New York, 2004, pp. 217–226.

A. Doan, N.F. Noy, and A.Y. Halevy, "Introduction to the special issue on semantic integration," *SIGMOD Record*, vol. 33, no. 4, 2004, pp. 11–13.

D. Fensel, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer, Berlin, 2003.

A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari, "Sweetening WordNet with DOLCE," *AI Magazine*, vol. 24, no. 4, 2003, pp. 13–24.

D. Gasevic, D. Djuric, and V. Devedzic, "Bridging MDA and OWL ontologies," *Journal of Web Engineering*, vol. 4, no. 2, 2005, pp. 118–143.

D. Gasevic, D. Djuric, and V. Devedzic, *Model Driven Architecture and Ontology Development*, Springer, Berlin, 2006.

C.H. Goh, *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems*. MIT, Cambridge, 1996.

A. Halevy, "Why your data won't mix," *Queue*, vol. 3, no. 8, 2005, pp. 50–58.

H.J. Happel, and S. Seedorf, "Applications of ontologies in software engineering," *2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2006.

M. Hepp, P. De Leenheer, A. De Moor, and Y. Sure, *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, Springer, Berlin, 2007.

J. Miller, and J. Mukerji, "Model driven architecture (MDA)," *Object Management Group, Draft Specification ormsc/2001-07-01, July*, vol. 9, 2001.

J. Miller, and J. Mukerji, "Model Driven Architecture ((MDA) Object Management Group Draft Specification," 2001; http://www.omg.org/docs/ormsc/01-07-01.pdf).

R. Mordinyi, T. Moser, E. Kühn, S. Biffl, and A. Mikula, "Foundations for a model-driven integration of business services in a safety-critical application domain," *35th Euromicro Conference Software Engineering and Advanced Applications (SEAA)*, 27–29 August 2009, IEEE Computer Society, Washington, 2009, pp. 267–274.

T. Moser, A. Andjomshoaa, and A. Mikula, "FISN Semantic Architecture Document," *Book FISN Semantic Architecture Document*, Series FISN Semantic Architecture Document, Frequentis AG, 2007.

T. Moser, R. Mordinyi, A. Mikula, and S. Biffl, "Efficient system integration using semantic requirements and capability models: an approach for integrating heterogeneous business services," *11th International Conference on Enterprise Information Systems (ICEIS 2009)*, 2009, pp. 56–63.

T. Moser, R. Mordinyi, W.D. Sunindyo, and S. Biffl, "Semantic service matchmaking in the ATM domain considering infrastructure capability constraints," *21st International Conference on Software Engineering and Knowledge Engineering*, 2009, pp. 222–227.

T. Moser, K. Schimper, R. Mordinyi, and A. Anjomshoaa, "SAMOA—A semi-automated ontology alignment method for systems integration in safety-critical environments," *2nd IEEE International Workshop on Ontology Alignment and Visualization (OnAV'09)*, 2009, pp. 724–729.

I. Niles, and A. Pease, "Towards a standard upper ontology," *2nd International Conference on Formal Ontology in Information Systems*, ACM, New York, 2001, pp. 2–9.

N.F. Noy, "Semantic integration: a survey of ontology-based approaches," *SIGMOD Record*, vol. 33, no. 4, 2004, pp. 65–70.

N.F. Noy, A.H. Doan, and A.Y. Halevy, "Semantic integration," *AI Magazine*, vol. 26, no. 1, 2005, pp. 7–10.

J. Pollock, "Integration's dirty little secret: it's a matter of semantics," *Whitepaper, Modulant: The Interoperability Company*, 2002.

M. Uschold, and M. Gruninger, "Ontologies and semantics for seamless connectivity," *SIGMOD Record*, vol. 33, no. 4, 2004, pp. 58–64.

H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information-a survey of existing approaches," *Workshop on Ontologies and Information Sharing (IJCAI-01)*, 2001, pp. 108–117.

Chapter 2
An Ontology-Based Approach for Supporting Business-IT Alignment

Csaba Veres, Jennifer Sampson, Karl Cox, Steven Bleistein, and June Verner

Summary B-SCP (Business Strategy, Context, and Process) is a promising framework addressing alignment of IT with business strategy from a requirements engineering perspective. The B-SCP approach combines goal and context modeling, and business processes, into a generic modeling framework that deconstructs these to IT requirements and context. However, a problem with the B-SCP framework is that it is difficult to track dependencies between requirements in a project of realistic complexity. To address this we discuss how the RDF (Resource Description Framework) data model with OWL (Web Ontology Language) semantics will greatly benefit an implementation using B-SCP. Our contribution is to extend B-SCP by describing an ontology data structure for representing the requirements and the complex rules which map them together. The benefit in our approach is that it provides a comprehensive way to validate the decomposition of the requirements. Seven–Eleven Japan is used as an exemplar to demonstrate improved productivity and consistency of B-SCP.

C. Veres (✉)
University of Bergen, Bergen, Norway
e-mail: Csaba.Veres@infomedia.uib.no

J. Sampson
Statoil, Bergen, Norway
e-mail: jensam@statoil.com

K. Cox
University of Brighton, Brighton, UK
e-mail: k.cox@brighton.ac.uk

S. Bleistein
Enterprise Analysts Pty Ltd, Sydney, Australia
e-mail: steve@enterpriseanalysts.com

J. Verner
University of New South Wales, Sydney, Australia
e-mail: june.verner@gmail.com

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_2,

2.1 Introduction

Sustainable information systems are information systems that not only meet current IT business requirements, but also continue to meet changing IT requirements and goals. We argue that it is imperative at the outset of the project, to adopt an approach that facilitates the development and maintenance of sustainable information systems. To that end we will describe an extension to B-SCP, a promising framework addressing alignment of IT with business strategy from a requirements engineering perspective. The approach we describe is particularly relevant for medium to large enterprises with great depth and complexity.

There is an increasing amount of research devoted to utilizing semantic web technologies in software engineering and requirements engineering, e.g. (Dobson and Sawyer 2006; Mayank et al. 2004; Yu and Mylopoulos 1994). We contribute to this effort by presenting a knowledge management approach for managing requirements. That is, we enhance the requirements engineering framework B-SCP by implementing an ontology data structure for representing the requirements and the complex rules aligning them.

First, in Sect. 2.2 we briefly review B-SCP, an integrated requirements engineering framework that enables verification and validation of requirements in terms of alignment with, and support for, business strategy. Then in Sect. 2.3 we discuss some related work. In Sect. 2.2.3 we describe the Seven–Eleven Japan exemplar which we use throughout the paper to illustrate our approach. Next in Sect. 2.4, we describe how we use the RDF data model with OWL semantics to create a Business Motivation Ontology for the B-SCP framework. In Sect. 2.4.2 we explain how we leverage a number of existing semantic technologies in a simple implementation of the framework. Following the presentation of our ontology, we briefly discuss our findings in Sect. 2.5. Finally, in Sects. 2.6 and 2.7, we describe limitations of our approach, future plans and conclude our work.

2.2 Background

2.2.1 What is Business-IT Alignment?

Business-IT alignment can be characterized as follows. IT can consistently be employed to strengthen or raise the performance of the business, and IT systems can be put to work in the business without employee ‘angst’. For many companies to be better aligned, however, aggressive actions will need to be taken to ensure the IT function is structurally aligned with the business, to ensure that the company as a whole is supportive in investments in IT assets, and to develop and implement methods and tools that allow systems implementations to be both technical and business successes (Chan and Reich 2007).

As far as we are aware, there are no methodologies that guide end-to-end business-IT alignment—from business strategy through to business processes and

requirements (though see our discussion on Related Work). The fact that it is extremely difficult to know if IT is aligned to strategy often leads to poor execution, cost overruns and failure to manage risk (Chan and Reich 2007).

Though there are many frameworks and methodologies for business-to-IT alignment (Chan and Reich 2007) almost none address elicitation or graphical modeling. One that does address elicitation (Sondhi 1999) proposes a simple set of six questions to elicit vision, mission, business strategy, goals, and objectives; this was not tied to any modeling framework. However, B-SCP is unique in that it utilizes Sondhi's questions (Bleistein et al. 2006) and places answers in a table so that a model can be constructed.

2.2.2 The B-SCP Framework

An integrated approach for organizational IT has been proposed by Bleistein (2006), Bleistein et al. (2005, 2006). The approach describes a "requirements analysis framework that enables verification and validation of requirements in terms of alignment with and support for business strategy". B-SCP combines goal and context modeling, and business processes, into a generic modeling framework that deconstructs these to IT requirements and context. The motivation is to address the "gap between requirements engineering and analysis approaches and frameworks for validating strategic alignment of organizational IT" through determining "existing requirements engineering techniques that might be applied to modeling and analysing business strategy" (Bleistein et al. 2006) (p. 363).

The power of the approach is that it combines problem diagrams (Jackson 2001) and requirements engineering $i*$ goal modeling (Anton 1996; Chung et al. 1999; Dardenne et al. 1993; Yu 1993; Liu and Yu 2001), to form in an integrated framework. Moreover, Bleistein introduces VMOST (Vision, Mission, Objective, Strategy, Tactic) analysis (Sondhi 1999), an organizational alignment technique for deconstructing business strategy into core components by answering a number of key questions. The process of using VMOST analysis, together with the BMM (Business Motivation Model) model to develop a goal model of organizational business strategy, is discussed in detail in Bleistein (2006), Bleistein et al. (2005, 2006). The authors describe how the core components can be used to construct a goal model of business strategy with guidance from the business motivation model (shown in Fig. 2.1).

According to the Business Rules Group (Healy and Ross 2007) BMM provides a "scheme or structure for developing, communicating, and managing business plans in an organized manner" (Healy and Ross 2007). That is, BMM provides the means to:

- identify factors that motivate the establishing of business plans;
- identify and define the elements of business plans;
- indicate how all these factors and elements inter-relate.

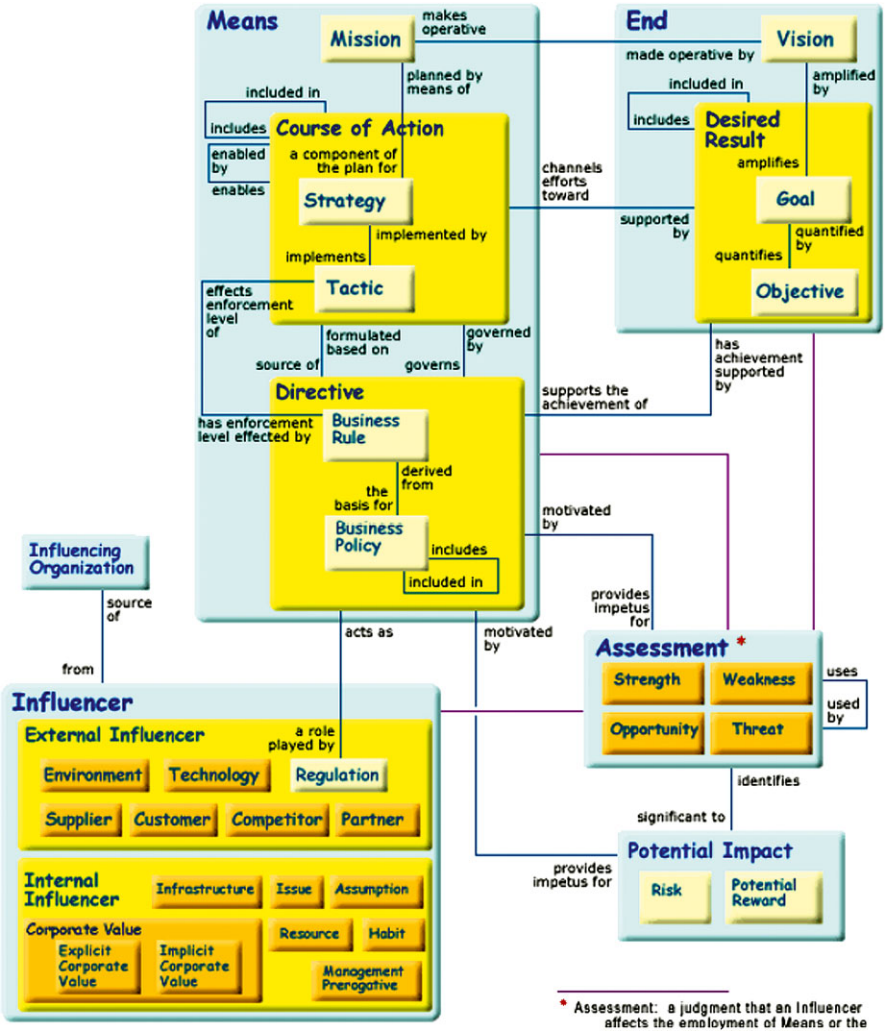


Fig. 2.1 Business motivation model (Healy and Ross 2007)

The major areas of the BMM are the *Ends* and *Means* of business plans, the *Influencers* that shape the elements of the plans, and the *Assessments* on the ends and means. Each of these components act towards answering the following fundamental questions (Healy and Ross 2007):

- What is needed to achieve what the enterprise wishes to achieve?
- Why does each element of the business plan exist?

Naturally, all elements of the BMM are to be developed from a business perspective. One outcome of applying BMM is a business model that captures the elements of business plans before system design or technical development commences. Overall

the Business Motivation Model acts as a blueprint purposely designed to support a range of methodological approaches (Healy and Ross 2007). Implementing the Model results in the elements of business plans being represented, captured and related to other information about the enterprise.

In the next section we will describe an exemplar to illustrate the connection between the BMM and the B-SCP approach.

2.2.3 Seven-Eleven Japan Example

Bleistein (2006), Bleistein et al. (2005, 2006) use the case of Seven–Eleven Japan (SEJ) and its IT system to illustrate the B-SCP approach. The literature describing SEJ (Bensaou 1997; Kilcrease et al. 1997; Makino and Suzuki 1997; Rapp 2002; Weill and Vitale 2001; Whang et al. 1997) provides a rich picture of both SEJ's business strategy and the IT system SEJ uses to implement its strategy to compete with business rivals. SEJ is succinctly summarized by Bleistein (2006),

> *"SEJ manages a national franchise of independently owned convenience stores. SEJ uses its IT to actively collect and analyse individual customer purchase pattern data at the point-of-sale in each franchise store, which are correlated with local social and environmental factors to develop a remarkably reliable, real-time predictive model. SEJ's IT system enables franchisees to predict customer purchasing behavior, store-by-store, item-by-item, hour-by-hour, effectively enabling management of a supply chain of business partners to stock stores just-in-time according to changing customer demand".*

We briefly describe how Bleistein et al. apply B-SCP to the SEJ case. First they present the results of VMOST analysis (Sondhi 1999) of SEJ's strategy using the following key questions from Sondhi (1999):

1. What is the overall, ideal, end-state toward which the organization strives (vision)?
2. What is the primary activity that the organization performs to achieve the end-state (mission)?
3. How are the responses to Questions 1 and 2 (vision and mission, respectively) appropriate and relevant to the environment?
4. Are the responses to Questions 1 and 2 (vision and mission, respectively) explicit or implied? How?
5. What are the basic activities and their rationale by which the organization competes with industry rivals?
6. What goals does the organization set to determine if it is competing successfully?
7. What activities does the organization perform to achieve the goals in Question 6?
8. How do the goals in Question 6 support the response to Question 1 (vision)?
9. What are the measurable objectives that indicate achievement of goals identified in Question 6, and what activities does the organization perform to achieve those objectives?

10. How do the objectives identified in Question 9 support the goals identified in Question 6?

Answers to these questions are illustrated using VMOST analysis for Seven–Eleven Japan mapped to a BMM model in Fig. 2.2. Rules governing contribution relationships according to the BMM model are captured in Table 2.1. Involved domains of interest are identified as the first step in constructing context diagrams and detailing shared phenomena, requirements references and constraints. In B-SCP all this information is necessary to integrate the goal model with the progression of problem diagrams. A goal model is then created using the components in Table 2.1. In the left-hand side of Fig. 2.2 a goal model is constructed from Table 2.1. The identified domains from Table 2.1 are used to help construct context diagrams that appear on the right-hand side of Fig. 2.2. The integration of the goal model and the context diagram at each level in the progression presents a problem diagram for that particular level of abstraction. The interested reader may read in detail the process of constructing the models in Bleistein (2006), Bleistein et al. (2005, 2006).

The business model and strategy is the top-level requirements engineering problem for SEJ, described by `Requirements_Set: RA`, and `Domain_Context: DA` in Fig. 2.2.

Requirements may impose a `Constraint` on a `Domain_of_Interest`, such a constraint is indicated by a dashed line with an arrowhead in Fig. 2.2. In each Context diagram on the right-hand side of Fig. 2.2 `Shared_Phenomena` are represented by lines between two domains. For example, in `Domain_Context: DA`, 'Consumer' and 'Franchise_Store' have `Shared_Phenomena` "Provision of products for purchase that consumers want, when they want them" represented by `a`.

2.3 Related Work

There are many approaches to combining business and software models. Two widely accepted approaches are extending UML into the business world and pushing model driven architecture into the problem domain. We consider both approaches below. Motivated by a desire of "software managers" and "developers" to be able to understand and communicate about the needs of a business in order to identify "proper requirements" before "producing [a] software system", Eriksson and Penker (2000) propose an extended version of UML in 'Business Modeling with UML' for the express purpose of modeling a business, rather than only its software. Eriksson and Penker present views of business vision, business structure, business process, and attached to business process, business behavior. The term "Strategy definition" is used in the business vision view (Eriksson and Penker 2000). On the face of it, these views appear to be eminently sensible as a first step in modeling a business in order to achieve business-IT alignment.

To support these views, they propose a set of UML extensions for business modeling, including goal modeling for the business vision view and process modeling as an "assembly line" for the business process view (Eriksson and Penker 2000).

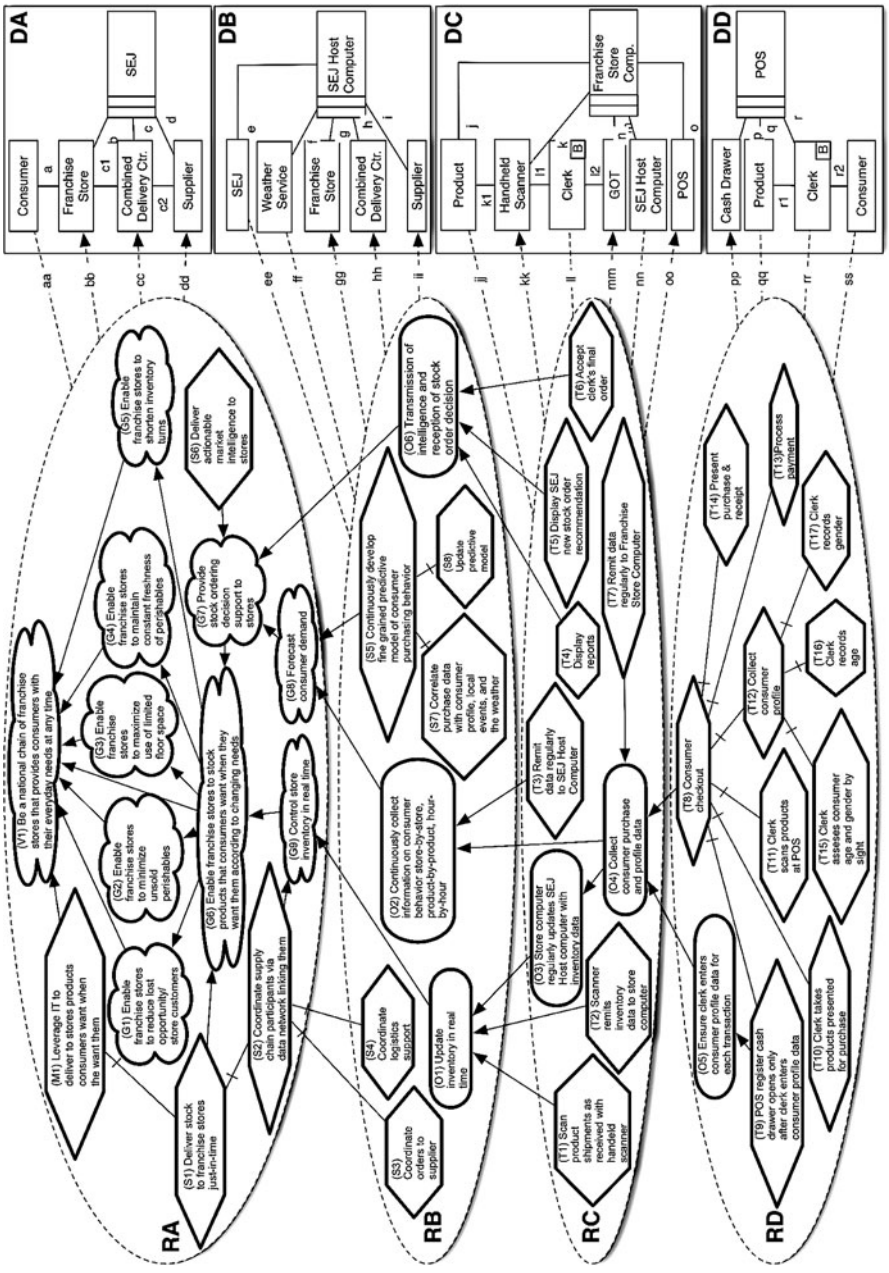


Fig. 2.2 SEJ integrated goal model and context diagrams in progression of problems (Bleistein 2006)

Table 2.1 VMOST to BMM (BRG-Model) mapping (adapted from Bleistein et al. 2006; Bleistein 2006)

ENDS				MEANS			
ID	BMM Entity type	Links To	Involved Domains	ID	BMM Entity type	Links To	Involved Domains
	Vision(Soft Goal)				**Mission (Task)**		
V1	Be a national chain of franchise stores that provide consumers with their everyday needs at any time.		Consumer, Franchise Store SEJ	M1	Leverage IT to deliver to stores products consumers want when they want them.	V1	Consumer Franchise Store SEJ
	Goals (Soft Goals)				**Strategies (Tasks)**		
G1	Enable franchise stores to reduce lost opportunity/store consumers.	V1	Consumer, Franchise Store, SEJ	S1	Deliver stock to franchise store just-in-time	M1	Franchise store, SEJ, Supplier Combined Delivery ctr.
G2	Enable franchise stores unsold perishables	V1	Franchise store SEJ	S2	Coordinate supply chain participants via data network linking them	S1, G9	Franchise store, SEJ, Supplier, Combined Delivery ctr.
G3	Enable Franchise stores to maximize use of limited floor space	V1	Franchise store SEJ	S3	Coordinate orders to supplier	S2	Franchise store SEJ Host Computer, Supplier
G4	Enable Franchise stores to maintain	V1	Franchise store SEJ	S4	Coordinate logistics support	S2	Franchise store, SEJ Host Computer, Supplier, Combined Delivery ctr.
G5	Enable Franchise stores to shorten inventory turns	V1	Franchise store SEJ	S5	Continuously develop fine grained predictive model of consumer behavior	G8	SEJ, Weather Service, Franchise Store, SEJ Host Computer
G6	Enable Franchise stores to stock products that consumers want when they want them	V1, G1-6	Consumer, Franchise store, SEJ	S6	Deliver actionable market intelligence	G7	SEJ, Franchise Store, SEJ Host Computer
G7	Provide stock ordering decision support to stores	G6	Franchise store, SEJ	S7	Correlate purchase data with consumer profile, local events	S5	Weather service SEJ Host Computer
G8	Forecast consumer demand	G7	Franchise store, SEJ	S8	Update predictive model	S5	SEJ, SEJ Host computer
G9	Control store inventory	G8	Franchise store SEJ, Supplier				

The extension for goal modeling is simple, consisting of only one goal type and one contribution relationship type. The UML extension for process modeling treats processes as a set of ordered activities that add value to deliver an output to a market or customer, like an "assembly line process" (Eriksson and Penker 2000). No extension is offered for business structure, which is performed using standard UML class diagrams (Object Management Group 2004) to represent organizational structures and relationships among participants in a business model.

While Eriksson and Penker (2000) provide no example of defining and modeling business strategy, their approach has been applied to modeling "business strategy" in information systems research and is presented in case studies appearing in Mendes et al. (2001), Vasconcelos et al. (2001). However, these case studies are taken from a business process modeling case in the literature and only demonstrate the process modeling aspects, failing completely to address strategy modeling as claimed. Eriksson and Penker (2000) assert that the primary advantage of their approach is that it is UML and easy to pick up for those already familiar with UML. However UML, whose purpose is primarily that of modeling software design, has a number of shortcomings when used for business modeling. First, those who are familiar with UML are designers, systems analysts and business analysts. Business managers would not have the first clue about UML. Second, the UML goal modeling extension lacks the richness of other established requirements engineering goal modeling notations and frameworks such as KAOS (Dardenne et al. 1993), i* (Yu 1993), and GBRAM (Anton 1996). Third, the "assembly line process" modeling proposed by Eriksson and Penker (2000) focuses on process flow, which unfortunately makes it an awkward notation for understanding business processes that do not match the "assembly line" pattern of bundles of work passing through manufacturing-type processes to build a product. Fourth, modeling business structure using UML class diagrams is awkward. In the real world of an organization, do departments of a firm inherit attributes from some common super class of departments? Are the attributes of a clerk, such as name, employee ID, date hired, relevant, i.e., useful and necessary, to the contextual domain of all problems in which the clerk appears? In a "real" ATM system, does a "cash note" know its attribute of being either "on hand" or "dispensed," or know its value, as suggested by Kaindl (2005)? Such concepts, which are common to object-oriented software design, bear little resemblance to the context of the real world and are at times, as in the case of Kaindl (2005), somewhat absurd. Indeed, UML has been recognized as an inappropriate notation for modeling domain context of the real world (Evans 2004; Jackson 1995, 2001; Kovitz 1999; Robertson and Robertson 1994).

In addition, Business modeling with UML has a number of major shortcomings when used for business-IT alignment in requirements analysis. First, Eriksson and Penker (2000) do not demonstrate how the "business views" connect and integrate with each other. Business vision, structure, and process are modeled independently of each other and have no explicit cross-referencing mechanism to help verify alignment of views with each other. For example, if there is a change in the organizational structure, there is no mechanism for tracing how that change might impact organizational goals or business processes. Second, as Eriksson and Penker (2000) state, the UML software architecture model is distinct from the UML business model, and thus there is no means to explicitly trace between the business model and system requirements. This separation of business model from system model severely limits the capacity of using Business modeling with UML to provide explicit requirements traceability to business strategy. Third, Business modeling with UML appears to restrict its views primarily to what is internal to the enterprise. It is not clear whether this is a result of a design limitation of Business modeling with UML,

or rather simply a result of the nature of the case examples presented in Eriksson and Penker (2000), Vasconcelos et al. (2001) which focus heavily on internal, operational concerns, with little reference to the external environment. As business strategy concerns primarily what is outside the enterprise (Hamel and Prahalad 1994; Mintzberg et al. 1998; Oliver 2001; Porter 1996; Porter and Millar 1985; Quinn et al. 1988), such as competitors, customers, suppliers, and business partners, the case examples simply do not address the scope of business strategy. Limiting scope to internal operational concerns is a perilous practice for those who intend to perform business analysis for systems of strategic import.

Similarly, Model Driven Architecture (MDA) (Object Management Group 2003; Kleppe et al. 2003; Mellor et al. 2004) provides a framework for the development and maintenance of software systems that allows an analyst to describe both business and software assets, though, as Jeary et al. (2008) explain, MDA is heavily weighted in favor of software assets with almost no consideration of a sensible approach to modeling a business. Though the MDA does propose what it calls the Computation Independent Model (CIM) to cover aspects of business and business requirements, how this is actually done is not explained by the OMG (Object Management Group 2003). There appears to be no model type defined that covers requirements and specification other than text. Roles or users in the CIM space are also undefined. The EU 6th Framework Project: VIDE, (VIDE project 2007) describes a potential CIM role as an end user of the completed system, but there is no assertion that role is expected to have any knowledge of modeling, such as the Business Process Modeling Notation (Object Management Group 2008). Though BPMN has been proposed as a potential CIM modeling tool (Rungworawut et al. 2007), Jeary et al. (2008) report that BPMN is not an appropriate model for business managers because it is overly complex with many modeling elements not appropriate to the business domain. We also question the appropriateness of BPMN because in its OMG form it does not have a 'goal' element to indicate whether a goal has been achieved either completely or partially, hence making traceability to a strategy model very difficult. Given that a business process should achieve or help achieve a number of goals, we are convinced the BPMN in its current form is not suitable for modeling strategic alignment. However, others are beginning to address this, for example, Whitestein Technologies Goal-Oriented BPMN (Whitestein Technologies 2009).

In order to close the gap between the CIM level and business, Jeary et al. (2008) propose a "pre-CIM" layer, above the CIM, that will capture knowledge about "organizational hierarchies, informal documentation, private process views [e.g. internal processes that companies perform as part of a supply chain], details of responsibilities, order forms etc." Examples of what they mean are provided in Kanyaru et al. (2008), where a scrapbook concept is shown in capturing 'potential customer sales leads' such as can be recorded in Salesforce.com. This appears to be a shoot-from-the-hip or flying-blind strategy that most businesses try to avoid, except in sales. A sales strategy is not a business strategy. Jeary et al. propose the use of BPMN in the CIM layer as a means of formalizing the 'scrap book' scenarios in their pre-CIM. Though a business process should achieve a business goal and a collection

of processes should achieve a collection of business goals, there is no mention of a business goal model or, as we propose, a strategic business model that is an integrated framework of strategic, business and user goals and requirements within a business and project context. A business process model alone will not inform where the goal came from that it achieved nor the consequence of achievement upon other goals in a goal hierarchy.

In its current form, MDA researchers and practitioners have focused far more on the systems modeling with little more than a passing consideration for the business model. BPMN is the de facto standard for business process modeling and as such is the next logical step up the chain, especially when one considers that BPMN models can ultimately become executables. Jeary et al. and Kanyaru et al. have recognized the shortfall in MDA and have proposed useful solutions to those. However, they do not formally model a business strategy nor a business context, and these are not integrated. It is unclear how pre-CIM could model a change in a business strategy and demonstrate its impact upon a specification for the system.

2.4 Objectives and Contributions

The B-SCP framework defines a mapping of system requirements against strategic business objectives. In this chapter we now extend Bleistein's et al.'s previous research by precisely defining the semantics for each link in the mapping from VMOST to the BMM model, shown in Fig. 2.1. The actual mapping VMOST to the BMM is shown in Table 2.1. The semantically enriched model will allow us to make inferences about the chain of requirements and expose their properties and potential hidden assumptions which would otherwise be difficult to see.

2.4.1 Adding Semantics to the B-SCP Framework

The technology we use for semantic enrichment is Ontologies (Staab and Studer 2004), a cornerstone technology of the Semantic Web as coined by Tim Berners-Lee and colleagues (Berners-Lee et al. 2001). Ontologies are (typically) a logic-based data model which define the concepts and relations in a domain of interest. These concepts and relations have instantiations in a particular instance model, and the ontology can be used to infer interesting logical facts about the instances. One useful feature of ontology-based models is their ability to perform automatic classification. For example, consider a simple ontology in which we have the primitive concepts PERSON (subclassed to MALE and FEMALE) and the relation hasChild which holds between a PERSON and another PERSON. Further, we define (through the logical vocabulary) the concept FATHER, which is "a MALE who has at least one PERSON in a hasChild relation". Now suppose we have a number of instances of each concept:

- MALE: John, Jack, Peter, Henry, George;
- FEMALE: Helen, Destiny, June, Violeta;
- hasChild: (John, Destiny), (Peter, Helen).

If we now ask for all the FATHERs in the system, a reasoner can return John and Peter as result. That is, by defining complex concepts from primitive ones already in the system, it is possible to compose interesting views of the data in a system. For a slightly more interesting example, suppose that you were a very bright graduate student visiting a new university department and that you wanted to take an interesting class. If facts about classes, faculty, and students were held in an ontology-based system, you could define a concept INTERESTING-CLASS as one in which there were "fewer than 15 STUDENTS, all of whom have had an A or A+ in all of their previous CLASSES, and taught by a PROFESSOR who has at least 10 PUBLICATIONS a year". Most importantly, this query can be defined without any knowledge of the data source which contains the departmental information. The ontology of basic concepts provides sufficient semantics for the novel queries. We will see that such classification can be used effectively when trying to position requirements at various levels of organizational objectives.

Another important function that ontologies can provide is the detection of inconsistency. In other words, are there concepts defined in an ontology that simply cannot be instantiated? Are there contradictions in the definitions of complex concepts? These inconsistencies can be automatically inferred, which is of tremendous benefit in complex models where the interdependency of different concepts can become confused. Finally, ontologies provide an application free store of data. While this is partially possible with standard database technologies, the problem with these tends to be that databases are typically designed with particular applications in mind, and the meaning of each data point can only be understood with respect to the original application. Likewise, XML while very flexible the elements and attributes have no meaning by themselves, this means that it becomes difficult to determine the impact of a change to, for example, the order of tags or a change in the terminology. Ontologies on the other hand have an application-independent logical vocabulary for describing the data. The ontology concepts and relations can be exploited equally by any application, making knowledge sharing between applications possible. Thus, the products of a requirements engineering session can be used directly in an architecture design tool, or even at high-level strategic meetings. The role of ontologies in knowledge sharing and reuse is noted in Abecker and van Elst (2004).

2.4.2 Why an Ontology Based Approach?

One problem with the framework is that tracking dependencies between requirements becomes extremely difficult with any project of realistic complexity. Software support is clearly necessary. A dedicated application with useful graphical depictions of model concepts and their relationships is probably the preferred course. But

any such application will need suitable data structures that can support all the necessary queries and inferences that users might need. While some application specific implementations might be suitable, we suggest that the RDF data model with OWL semantics will greatly benefit the implementation for several reasons:

1. By using OWL/RDF, it is possible to leverage a number of existing technologies in the application. For example, one might want to incorporate the open-source SESAME or Jena for storage and inference. Such application independent stores will also allow the data to be easily shared with other applications that are SESAME aware, for example. Or, it might be possible to use existing DL reasoners to check for the consistency of requirements, goals, and so on. Implementing such reasoning in an application-specific manager would be extremely time consuming.
2. The level of tool support for semantic technologies is excellent and under constant development. For example, the application-independent data model allows other tools like Protégé[1] to browse and adjust the mapping data before a dedicated application is operational. An application-independent data model will also allow the development of queries (e.g. SPARQL) that might become necessary for the operations to be supported by the application.
3. Because the concepts in the models will have application-independent fixed semantics, they can be reused in other applications which might benefit from the semantics. For example, if some parts of the IT system that is being modeled are eventually constructed using a Service-Oriented Architecture, then the semantic descriptions of the original requirements and shared phenomena will be useful in describing the services themselves. In addition, the potential for reuse of other requirement-sets through cross-referencing is enhanced due to the ubiquity of the technology. A requirements engineer could potentially have access to a number of reusable ontologies, which are compatible with each other.

The remainder of this paper reports on our efforts to date in implementing a suitable ontology, partially shown in Fig. 2.3, for the framework, and experiments in the useful inference tasks that are made possible. In addition we continue to use Seven–Eleven Japan (SEJ) (Bleistein 2006) as an exemplar to show how we provide a precise, semiautomated way to manage the requirements models and the complex rules linking them.

2.4.3 Business Motivation Model Ontology

We formalize the mapping from VMOST to the BMM model using a ontology-based approach. The business motivation model proposed by the Business Rules Group is mapped to an ontology partially shown in Fig. 2.3. Individual requirements are mapped as individuals (or instances) of the ontology (not shown in the figure).

[1] http://protege.stanford.edu.

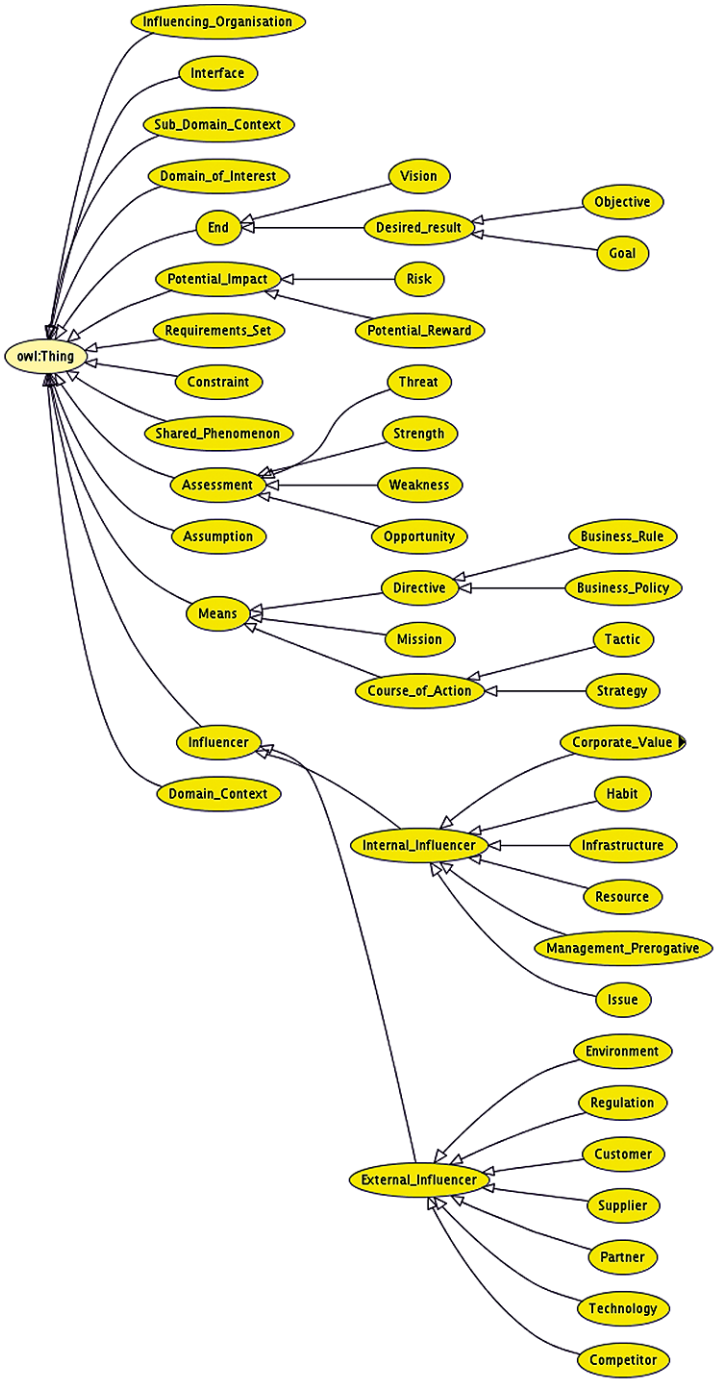


Fig. 2.3 Business motivation model class hierarchy

All of the relations shown in Fig. 2.1 are defined using properties in the Web Ontology Language (OWL). In the original framework the mapping was referred to as 'links to' (refer Table 2.1) for all requirement types related to: Vision, Mission, Strategy, Objectives and Tactics. However, from the Business Motivation Model we see that the semantics of the relations are not all the same. By using an ontology to represent the Business Motivation Model and Bleistein's et al's VMOST mapping we help manage and support the alignment of IT requirements with business strategy. In addition, the benefit of defining such semantics is that we can take advantage of many of the tools available for knowledge management such as reasoners and rule engines. For example, in the future we can use a combination of backward chaining reasoners (e.g. Jess) and DL subsumption reasoning to prove requirements decomposition.

Tracing the links between models in the SEJ-integrated goal model and context diagrams, shown in Fig. 2.2, is relatively difficult to follow manually. Our contribution is to use an ontology approach to manage the requirements models (goal model and context diagram combined) and the mappings between them. Obviously when describing very simple examples of requirements models, it is relatively easy to show the links between models. However in reality it is likely that an organization may have hundreds of requirements specifications and complex rules linking them together. Furthermore, the original approach does not address changing requirements according to changing strategy over time. Incorporating an ontology-based approach to represent requirements means we can take advantage of reasoning tools to help understand the impact of changing requirements.

In addition, our extensions help make the original B-SCP framework applicable throughout the lifecycle of a project instead of just at the requirements engineering phase. The knowledge repository will play an important role in not only capturing high-level business IT requirements but may eventually include additional constructs such as business rules, business policies and means for assessment. The BMM includes many other constructs not applied in the original B-SCP framework, because B-SCP was specifically focused around an explicit analysis of business strategy via (VMOST and BMM) to the requirements analysis frameworks, problem diagrams and goal-modeling.

2.5 Discussion and Results

Now that we have developed an ontology knowledge base we can write (SQL like) SPARQL queries against the complete model. For example, we can ask questions like "what are all the requirements in context D that serve vision x?", or "what tactics implement each Goal?". We performed a number of simple queries, we will only discuss three briefly to demonstrate the kinds of questions that might be asked. (Results of the first two queries can be found in Fig. 2.4 and Fig. 2.5.)

Query 1) What are the requirements for each requirement set?
PREFIX re:< http://www.semanticweb.org/ontologies/2007/10/RE.owl# >

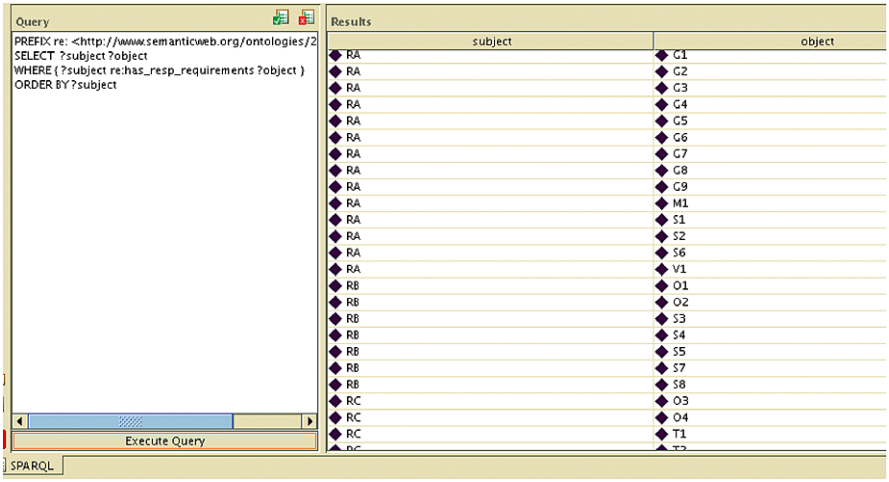


Fig. 2.4 Query 1: Requirements for each set (RA, RB, RC, RD1, RD2)

SELECT ?x ?z
WHERE {?x re:has_requirements ?z}

Query 2) The first query can be refined to: "find all the requirements of the requirements set RA":
PREFIX re: < *h*ttp://www.semanticweb.org/ontologies/2007/10/RE.owl# >
SELECT ?z
WHERE {re:RA re:has_requirements ?z}

Query 3) Next we chain queries to track Vision though goals:
PREFIX re: <http://www.semanticweb.org/ontologies/2007/10/RE.owl# >
SELECT ?z ?y ?a ?b
WHERE
{re:V1 re:has_goal ?z .
?z re:goal_description ?y .
?z re:has_sub_goal ?a .
?a re:goal_description ?b}

Using an ontology-based approach means we are able to leverage existing query tools to ask questions of our knowledge base instead manually checking the models. In Fig. 2.6 we see the results for Query 3. It shows that even though vision V1 has three separate goals G1, G2 and G3, these in turn all have the same subgoal, G6. So the primary underlying goal of the SEJ vision is to "Enable franchise stores to reduce scrap rates".

In Fig. 2.7 we show our BMM ontology using Protégé. In this multiview screen shot, we see instance DC of concept 'Domain_Context' with associated object and

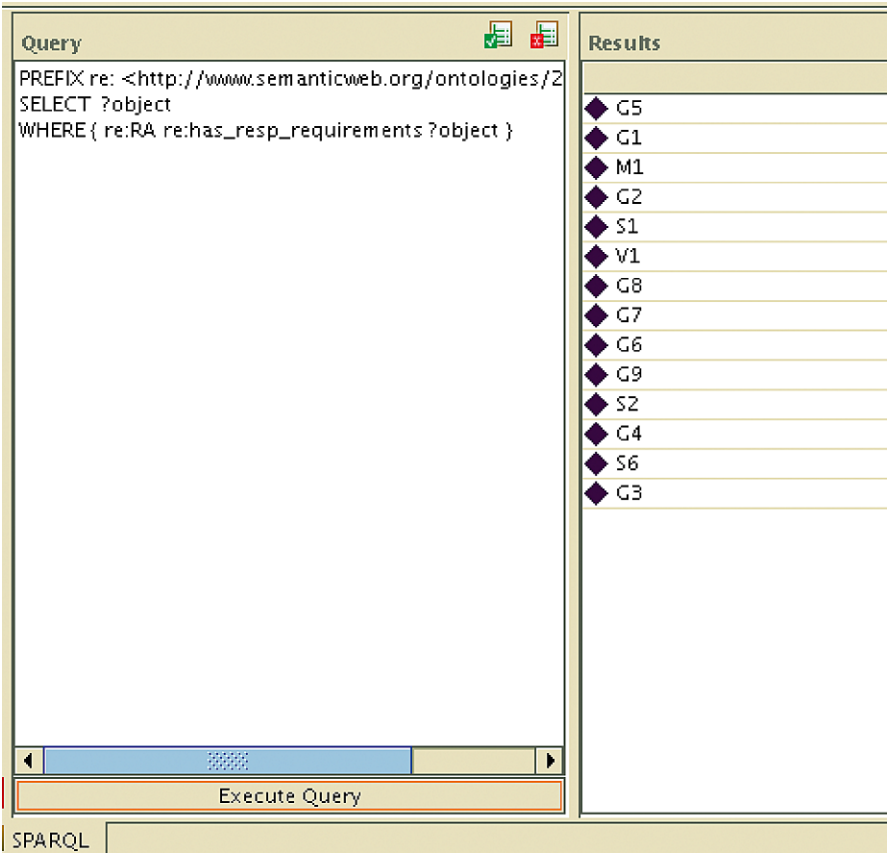


Fig. 2.5 Query 2: Refined query—requirements for set 'RA')

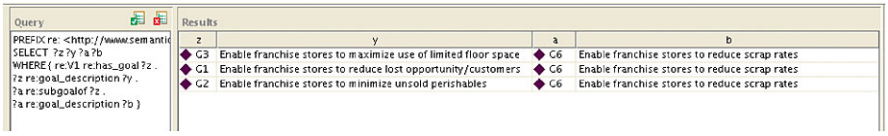


Fig. 2.6 Query 3: Tracking *Vision* through *Goals*

data property assertions is highlighted. Viewing the RDF data model using knowledge management tools such as Protégé allows a user, or requirements engineer, to analyse key relationships and links through each layer of the requirements models.

In a recent empirical study, Estrada et al. (2006) describe a number of limitations with the $i*$ goal modeling approach. Amongst other features, Estrada et al. (2006) found that the $i*$ goal modeling approach did not support complexity management nor traceability well, from a modeling language perspective. In the original B-SCP approach, complexity management and traceability were also problematic due to

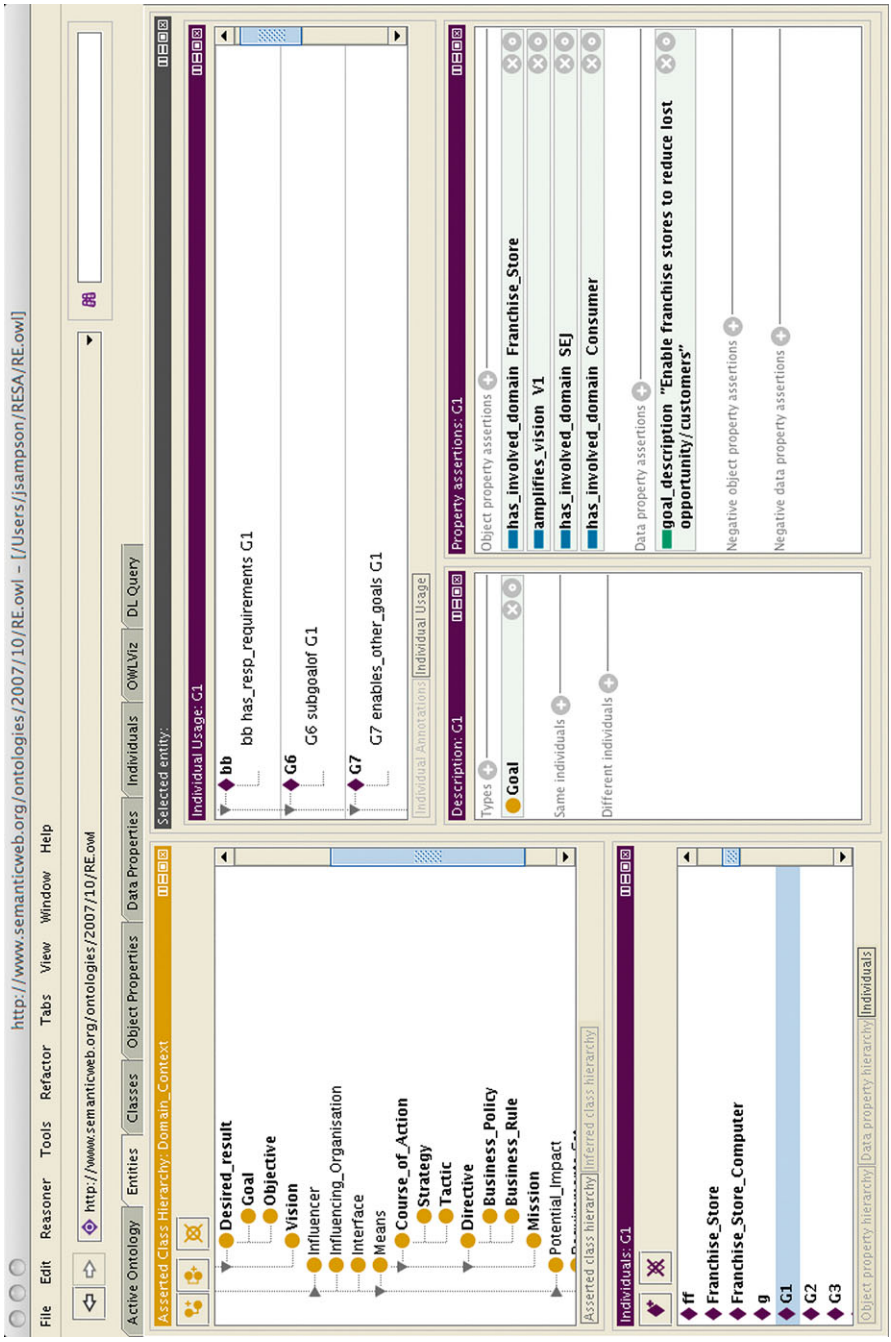


Fig. 2.7 BMMO with SEJ instances, properties and instance usage view

the use of $i*$ goal modeling approach and the complexity associated with model decomposition. However, since we have developed a Business Motivation Ontology, we have shown how we can productively manage complexity and traceability.

2.6 Future Work

One of the advantages of using the RDF data model with OWL semantics in our approach is that it provides a comprehensive way to validate the decomposition of the requirements. We intend to use a combination of backward chaining reasoners (e.g. Jess) and DL subsumption reasoning. In this way we have a way of proving each requirement set in the decomposition from the lowest level up to the Vision.

In team-based development, discipline specific models and trees of requirements may be developed by separate groups. To obtain a complete description of the requirements (and system architecture), these discipline specific viewpoints need to be brought into alignment. In previous research we developed a tool for representing ontology mappings in a visual way (Lanzenberger and Sampson 2006; Sampson and Lanzenberger 2006). By using such a tool we can visualize the differences between discipline specific viewpoints.

We also consider that VMOST analysis and the BMM model may not necessarily be appropriate in all cases. For example, an organization may not know what its strategy is or may not be able to articulate it (Bleistein 2006). In this case, both VMOST analysis and the BMM model may not be so useful, and other means for eliciting and analysing organizational motivation such as scenario authoring, by Rolland et al. (1998), may be more appropriate. Scenario authoring is a goal-oriented methodology for organizational IT that provides a linguistically based algorithm for developing variations and permutations on questions. Through this technique, it is possible to ultimately map out an organizations IT needs. Exploring other such elicitation methods will form part of our future work.

Next steps are to automatically populate the ontology with instances from the goal and problem diagrams. In the next phase of the research we will analyse a number of scenarios involving changing goals, and correspondingly how to represent the changes using our ontology. Providing a data structure to support changing requirements is an extremely important aspect of the approach as no systems are static.

2.7 Conclusion

We have presented a knowledge management approach for managing requirements. The benefits of using an ontology-based approach are: first, it becomes possible to leverage a number of existing technologies (like automated deduction) in the application. Second, the application-independent data model allows other tools like Protégé, to browse and adjust the data before the dedicated application is operational.

Third, it will also allow the development of queries (e.g. SPARQL) that might become necessary for the operations to be supported by the application. Fourth, since the concepts in the models will have application-independent fixed semantics, they can be reused in other applications which might benefit from the semantics.

We recognize that much work remains to be done to test the efficiency of the approach in industry. However, we have shown that SEJ is a useful exemplar as the number of requirements is not insignificant and tracing the links between the models is difficult without our ontology. It is our intention that the work be applied in an industrial setting.

Acknowledgement This work was conducted using the Protégé resource, which is supported by grant LM007885 from the United States National Library of Medicine.

References

A. Abecker and L. van Elst. Ontologies for knowledge management. In S. Staab and R. Studer, editors. *Handbook on Ontologies*. *International Handbooks on Information Systems*. Springer, Berlin, pages 435–454, 2004.

A. I. Anton. Goal-based requirements analysis. In *International Conference on Requirements Engineering*, pages 136–144. IEEE, New York, 1996.

M. Bensaou. *Seven–Eleven Japan: Managing a Networked Organization*. INSEAD, Euro–Asia Centre, 1997.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

S. J. Bleistein. *B-SCP: an integrated approach for validating alignment of organizational IT requirements with competitive business strategy*. PhD thesis, University of New South Wales, Sydney Australia, 2006.

S. J. Bleistein, K. Cox, and J. Verner. Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams. *The Journal of Systems and Software*, 2(79):362–378, 2006.

S. J. Bleistein, K. Cox, and J. Verner. Strategic alignment in requirements analysis for organizational it: an integrated approach. In *20th ACM Symposium on Applied Computing, track on Organisational Engineering ACM SAC 2005*, pages 1300–1307, Sante Fe, USA, March 2005.

Y. E. Chan and B. H. Reich. It alignment: what have we learned? *Journal of Information Technology*, 22(4):297–315, 2007.

L. E. Chung, B. Nixon, E. Yu, and J. Mylopoulos. *Non-functional Requirements in Software Engineering*. Kluwer Academic, Dordrecht, 1999.

A Dardenne, A. Van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science and Computer Programming 20*, pages 3–50, 1993.

G. Dobson and P. Sawyer. Revisiting ontology-based requirements engineering in the age of the semantic web. In *Dependable Requirements Engineering of Computerised Systems at NPPs*, Institute for Energy Technology (IFE), Halden, 2006.

H.-E. Eriksson and M. Penker. *Business Modeling with UML: Business Patterns at Work*. Wiley, New York, 2000.

H. Estrada, A. M. Rebollar, O. Pastor, and J. Mylopoulos. An empirical evaluation of the $i*$ framework in a model-based software generation environment. In *Proceedings of the 18th International Conference Advanced Information Systems Engineering (CAiSE 2006)*, pages 513–527, 2006.

E. Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley, Boston, 2004.

G. Hamel and C. K. Prahalad. *Competing for the Future*. Harvard Business School Press, Boston, 1994.

K. A. Healy and R. G. Ross. The business motivation model—business governance in a volatile world. Technical Report Release 1.3, The Business Rules Group, 2007.

Object Management Group. Mda guide version 1.0.1. Retrieved from http//www.omg.org/mda/specs.htm, 2003.

M. J. Jackson. *Software Requirements and Specifications: a Lexicon of Practice, Principles, and Prejudices*. ACM/Addison-Wesley, New York, 1995.

M. Jackson. *Problem Frames: Analyzing and Structuring Software Development Problem*. Addison-Wesley Reading, 2001.

S. Jeary, A. Fouad, and K. Phalp. Extending the model driven architecture with a pre-CIM level. In *TOOLS EUROPE Workshop, 1st International Workshop on Business Support and MDA (MDABIZ)*, Zurich, July 2008.

H. Kaindl. Is object-oriented requirements engineering of interest? *Requirements Engineering Journal*, 10(1):81–84, 2005.

J. Kanyaru, M. Coles, S. Jeary, and K. Phalp. Using visualisation to elicit domain information as part of the model driven architecture (MDA) approach. In *TOOLS EUROPE Workshop, 1st International Workshop on Business Support and MDA (MDABIZ)*, Zurich, July 2008.

D. P. Kilcrease, M. S. Murillo, L. A. Collins, and R. Kunitomo. Seven–Eleven is revolutionising grocery distribution in japan. *Long Range Planning*, 30(6):887–889, 1997.

A. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture—Practice and Promise*. Addison-Wesley Professional, Boston, 2003.

B. L. Kovitz. *Practical Software Requirements: a Manual of Content and Style*. Manning, Greenwich, 1999.

M. Lanzenberger and J. Sampson. Alviz—a tool for visual ontology alignment. In *Proceedings of International Symposium of Visualization of the Semantic Web, (IV06-VSW) 10th International Conference Information Visualization*. IEEE Computer Society, Los Alamitos, 2006.

L. Liu and E. Yu. From requirements to architectural design—using goals and scenarios. In *Proceedings of the ICSE-2001 (STRAW 2001)*, pages 251–263. Springer, Berlin, 2001.

N. Makino and T. Suzuki. Convenience stores and the information revolution. *Japan Echo*, 44:44–49, 1997.

V. Mayank, N. Kositsyna, and M. A. Austin. Requirements engineering and the semantic web: Part ii. representation, management and validation of requirements and system-level architectures. In *ISR Technical Report 2004-14*. University of Maryland, 2004.

S. J. Mellor, S. Kendall, A. Uhl, and D. Weise. *MDA Distilled: Principles of Model Driven Architecture*. Addision-Wesley, Reading, 2004.

R. Mendes, A. Vasconcelos, A. Caetano, J. Neves, P. Sinogas, and J. Tribolet. Representing business strategy through goal modeling. *International Conference on Enterprise Information Systems (ICEIS)*, pages 884–887, 2001.

H. Mintzberg, B. W. Ahlstrand, and J. Lampel. *Strategy Safari: a Guided Tour through the Wilds of Strategic Management*. Free Press, New York, 1998.

Object Management Group. *UML 2.0 Superstructure Specification*. Object Management Group, Needham, 2004.

Object Management Group. Business process modeling notation specification version 1.1. Retrieved May 2008, from http://www.omg.org/spec/BPMN/, 2008.

R. W. Oliver. What is strategy, anyway? *Journal of Business Strategy*, *November/December*, pages 7–10, 2001.

M. Porter. What is strategy? *Harvard Business Review*, 74(6):61–78, 1996.

M. Porter and V. Millar. How information gives you competitive advantage. *Harvard Business Review*, 63(4):149–160, 1985.

J. B. Quinn, H. Mintzberg, and R. M. James. *The Strategy Process: Concepts, Contexts, and Cases*. Prentice-Hall, Englewood Cliffs, 1988.

W.V. Rapp. Retailing: Ito–Yokado Seven–Eleven Japan. In *Information Technology Strategies: How Leading Firms Use IT to Gain an Advantage*. Oxford University Press, New York, 2002.

J. Robertson and S. Robertson. *Complete Systems Analysis*. Dorset House, New York, 1994.
C. Rolland, C. Souveyet, and C. Ben Achour. Guiding goal modeling using scenarios. *IEEE Transactions on Software Engineering*, (24):1055–1071, 1998.
W. Rungworawut, T. Senivongse, and K. Cox. Achieving managerial goals in business process component design using genetic algorithms. In *5th IEEE International Conference on Software Engineering Research, Management and Applications, SERA 2007*, Busan, Korea, 20–22 August 2007, pages 409–418, 2007.
J. Sampson and M. Lanzenberger. Visual ontology alignment for semantic web applications. In *1st International Workshop on Semantic Web Applications: Theory and Practice (SWAT 2006) ER2006 Workshop*. *LNCS*, Springer, Berlin, 2006.
R. Sondhi. *Total Strategy*. Airworthy Publications International Ltd, 1999.
S. Staab and R. Studer, editors. *Handbook on Ontologies*. *International Handbooks on Information Systems*. Springer, Berlin, 2004.
VIDE project. Standards, technological and research base for the vide project, project evaluation criteria and user requirements definition. Framework 6 EU Commission IST033606STP, 2007.
A. Vasconcelos, A. Caetano, J. Neves, P. Sinogas, R. Mendes, and J. M. Tribolet. A framework for modeling strategy, business processes and information systems. In *5th International Enterprise Distributed Object Computing Conference (EDOC 2001)*. IEEE Computer Society, Seattle, pages 69–80, 2001.
P. Weill and M. Vitale. *Place to Space: Moving to EBusiness Models*. Harvard Business School Publishing Corporation, Boston, 2001.
S. Whang, C. Koshijima, H. Saito, T. Ueda, and S. V. Horne. Seven–Eleven Japan (GS18), 1997.
Whitestein Technologies. Goal-oriented business process modeling notation. Retrieved July 2009, http//www.whitestein.com/go-bpmn, 2009.
E. Yu. Modeling organizations for information systems requirements engineering. In *Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 34–41. IEEE Computer Society Press, Los Alamitos, 1993.
E. Yu and J. Mylopoulos. Using goals, rules, and methods to support reasoning in business process reengineering. In *Business Process Reengineering, International Journal of Intelligent Systems in Accounting, Finance and Management*, pages 1–13, 1994.

Chapter 3
EPCIS-Based Supply Chain Event Management

**Christoph Goebel, Sergei Evdokimov,
Christoph Tribowski, and Oliver Günther**

Summary The coordination of assembly networks still represents a major challenge in today's business environment. We present a Radio Frequency Identification (RFID)-based inter-organizational system architecture that provides the technological basis for appropriate decision support. While mapping requirements in terms of information storage and exchange to technical system features, we consistently refer to the standards specified by the international industry consortium EPCglobal. In contrast to the pull-based architecture proposed by EPCglobal that is designed to retrieve and process historical data with a long lifetime, our system architecture follows a push approach. It allows for the propagation of relevant decision support information on past and future events with short validity. The EPCglobal event data specification is extended to include the required context information. A common protocol layer that interconnects supply chain stages is described in detail. The use of the protocol layer in connection with standardized formats for event and context data supports the interoperability of information systems used in different organizations and facilitates the integration of event-based applications into enterprise architectures. Using analytical methods, we evaluate the pull- and push-based ar-

C. Goebel (✉)
International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA 94703, USA
e-mail: goebel@icsi.berkeley.edu

S. Evdokimov · C. Tribowski · O. Günther
Humboldt-Universität zu Berlin, Spandauer Straße 1, 10178 Berlin, Germany

S. Evdokimov
e-mail: evdokimov@wiwi.hu-berlin.de

C. Tribowski
e-mail: christoph.tribowski@wiwi.hu-berlin.de

O. Günther
e-mail: guenther@wiwi.hu-berlin.de

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_3,

chitectures with respect to efficiency and reliability: the push-based architecture is shown to be particularly suitable for the realization of SCEM applications.

3.1 Introduction

Supply Chain Event Management (SCEM) systems are decision support systems that allow for monitoring, prioritizing, and reacting to events pertaining to the flow of goods in a supply chain (Otto 2003). A supply chain event can be any change of state with respect to the flow of goods, currency, or information. SCEM applications allow for the specification of rules, which can be applied to streams of events in order to identify those which are critical, i.e., events which call for immediate action in order to prevent financial loss. The business value of an SCEM application depends on the degree of supply chain visibility and the degree of freedom regarding possible ways to react to critical events; however, since the available options for all actions are as dynamic as the various states of supply chain operations, the logical design of SCEM systems can be very demanding.

Supply chain visibility refers to the amount of information available to the supply chain manager and can be characterized according to the following dimensions: detail, timeliness, and accuracy. Radio Frequency Identification (RFID) technology can be used for efficient tracking of materials as they move though the supply chain (Niederman et al. 2007). In contrast to the bar code, it allows for concurrent reading of item identification data without direct line of sight up to a certain read range (Michael and McCathie 2005). RFID is expected to increase supply chain visibility along all three dimensions mentioned. Since SCEM requires a high degree of supply chain visibility, the introduction of RFID into a supply chain could increase the business value of SCEM applications.

Standardization of a number of components that make up the architecture of event management solutions is on the way. Apart from protocols and data schemas designed to serve the purpose of receiving, accumulating, filtering, and reporting events pertaining to particular Electronic Product Codes (EPCs), the international industry consortium EPCglobal has specified Electronic Product Code Information Services (EPCIS) that should be responsible storing and exchanging these events across the supply chain (EPCglobal 2007). While EPC formats and RFID reader protocols have come a long way, EPCIS is still in an early stage of development.

Although the software industry is quick to offer products able to process EPC data, the development of value-generating business applications still lags behind. As long as real-world applications are rare, it is hard to justify the definition of a comprehensive standard. Little academic research on supply chain wide decision support systems based on auto ID technologies has been published so far. Chow et al. (2007) provided a schematic description of an interorganizational information system based on RFID that provides visibility of the processes taking place at a third-party logistics provider via a web front-end. Trappey et al. (2009) described an intelligent agent system that, among other things, supports real-time surveillance

of production progress. Although these authors have provided interesting starting points for the realization of interorganizational event-based applications, they do not go into technical details concerning the data formats and protocols required to realize these applications. Further research on interorganizational decision support systems based on auto ID technology is thus needed. In particular, it has to be determined which architectures suit which business applications. Since standardization plays a significant role in the design of interorganizational information systems, research on the appropriateness of the current standards proposed by EPCglobal is warranted.

Motivated by the knowledge gap identified above, we focus on three promising areas for further research:

- The specification of concrete business applications of event-based interorganizational supply chain management systems.
- The interoperability of the different components that need to be integrated in order to realize such systems.
- The intra- and interorganizational management of the EPC context data provided by different enterprise applications.
- The requirements analysis of EPCglobal's architecture for the supply chain-wide exchange of EPC-related data for SCEM applications.

We believe that without a specific requirements analysis, the degree of system interoperability cannot be assessed. The type of context data that needs to be managed and exchanged naturally depends on the application. Our approach thus consists of first putting the discussion about EPC-based material tracking into a concrete business context. To this end, we describe the challenges involved in coordinating decentralized make-to-order assembly networks. Our choice of the business context and application example tries to be as simple and general as possible. Thereafter, we derive technical requirements that need to be addressed by the architectural design, in particular with respect to interorganizational system interoperability. We present an approach to realize a two-layered interorganizational event-based architecture. In contrast to the components proposed by EPCglobal, our architecture follows a push approach for the dissemination of event data.

Our main contributions are the following:

- We describe a relevant business application of event-based systems in a multi-organizational context.
- Business requirements are mapped to technical system features while consistently referring to current EPCglobal specifications.
- We specify a tentative protocol layer that serves to integrate heterogeneous enterprise systems that exchange EPC context data.
- We develop quantitative evaluation criteria and compare the centralized EPCIS-based architecture proposed by EPCglobal with the decentralized EPCIS-based architecture proposed in the next section.

This work is structured in the following way. In Sect. 3.2, we introduce relevant components of the EPCglobal network. Section 3.3 outlines the business application

we focus on. In Sect. 3.4, the main ideas behind and some details of our proposed architecture are presented. A comparison of the developed and the centralized architectures, the quantitative evaluation, and its results are presented in Sect. 3.5. In Sect. 3.6, we discuss our findings. Section 3.7 concludes this chapter and outlines further research opportunities.

3.2 EPCglobal Network

The accuracy and detail of RFID data are expected to open up new and more efficient ways to manage the supply chain and reduce many of the inefficiencies plaguing today's businesses such as incorrect deliveries, shrinkage, and counterfeiting. Since production and distribution of physical goods are seldom in the hands of one organization and efficiency gains in supply chains often emanate from centralized supply chain control, it makes sense in most cases to share selected RFID data among supply chain participants.

For the cross-company exchange of RFID reader events, the Auto-ID Center and the industry consortium EPCglobal have specified a stack of specifications which is currently one of the predominant standardization efforts of the RFID community (Floerkemeier et al. 2007). In this section, we will sketch the specifications of the EPCglobal network that are most relevant for this work.

3.2.1 EPCglobal Architecture Framework

The EPCglobal architecture framework (EPCglobal 2009) describes a number of components required to realize a platform-independent system architecture for collecting, filtering, storing, and retrieving EPC-related data. EPCglobal does not define the system architecture that end users have to implement but defines interfaces that the components of end users' systems (hardware and software) may implement. These interfaces as well as these hardware and software roles are separated in Fig. 3.1.

EPCglobal has specified air interfaces for several tag types and reader protocols that serve to effectively read out EPC data in multitag, multireader environments.

The Application Level Events (ALE) specification defines how to request EPC data from readers so that it can be used as input for higher-level applications (EPCglobal 2008c).

The EPCIS Capturing Application has the context information about the data capturing process and supervises the lower elements in the EPCglobal architecture as described above. Its main task is to create an EPCIS event and store it in the EPCIS repository using the EPCIS Capture Interface.

To enable easy access to EPC-related data, EPCglobal described several dedicated services. The Object Name Service (ONS) standard specifies a hierarchical

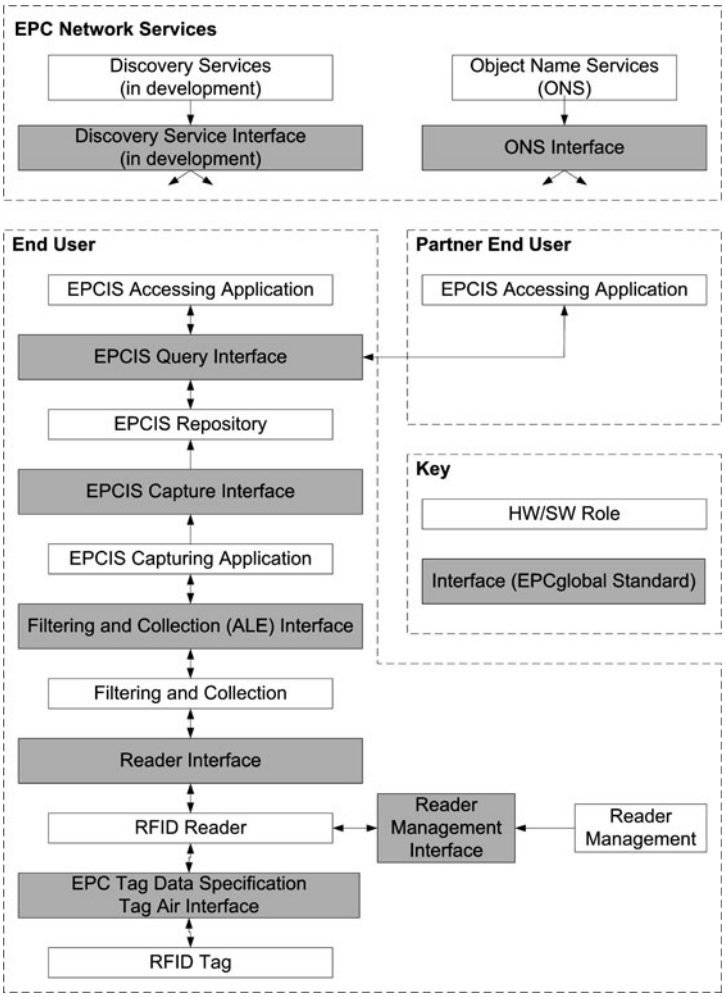


Fig. 3.1 EPCglobal architecture framework (based on EPCglobal 2009)

lookup service for locating service endpoints related to the EPC in question (EPCglobal 2008a). The EPC Discovery Services standard is currently under development, but the general description can be found in Kürschner et al. (2008). The main difference between the ONS and a Discovery Service is that EPC-related data provided by the services returned by the ONS is controlled by the entity that assigned the EPC to the item, while EPC-related data located using a Discovery Service can be controlled by any entity.

The EPC is described in more detail in Sect. 3.2.2, and EPCIS in Sect. 3.2.3.

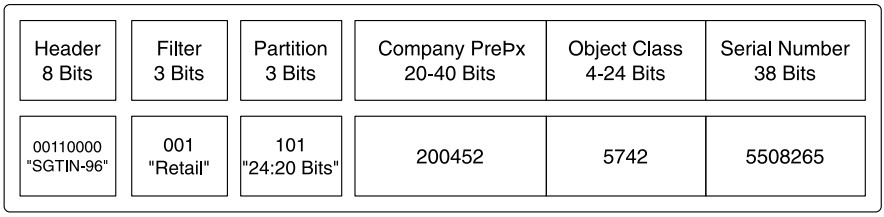


Fig. 3.2 EPC identifier example (SGTIN-96)

3.2.2 Electronic Product Code

The Electronic Product Code (EPC) specification describes a set of encoding schemas for universal identifiers that provide unique identities for physical objects (EPCglobal 2008b). The objects can be trade items (products), also logistical units, fixed assets, physical locations, etc.

The EPC schemas are designed to be backwards compatible with currently used GS1 codes. For example, the GS1 Serial Shipping Container Code (SSCC) for logistical units identifies unique objects. The corresponding EPC SSCC-96 encoding also identifies unique objects, and, therefore, there can be a one-to-one correspondence between SSCC and SSCC-96 identifiers. On the other hand, the Global Trade Item Number (GTIN)—the successor of the European Article Number (EAN)—identifies a category of objects, while the corresponding EPC Serialized GTIN (SGTIN-96) encoding allows identifying individual items by augmenting the GTIN with a serial number part.

Each EPCglobal subscriber manages its own range of EPCs contained within the organization's Company Prefix. The subscriber organization is responsible for maintaining the numbers of Object Classes and Serial Numbers which, together with the organization's Company Prefix, form the EPC (EPCglobal 2008b).[1] An example of such an EPC identifier encoded in SGTIN-96 format is displayed in Fig. 3.2.

3.2.3 EPC Information Services

The EPCIS, as conceived by EPCglobal, consists of three components: a repository for event data and two interfaces that serve to capture and query event data stored in this repository. Although EPCglobal does not provide an implementation of any of these components, they have developed the specification for an extendible data model for supply chain events as depicted in Fig. 3.3.

The capture interface receives formatted event data from the ALE and adds the required context data, resulting in one of the event types shown. The query interface

[1]The only exception is DoD-96 encoding that includes two parts and is used for shipping goods to the US Department of Defense.

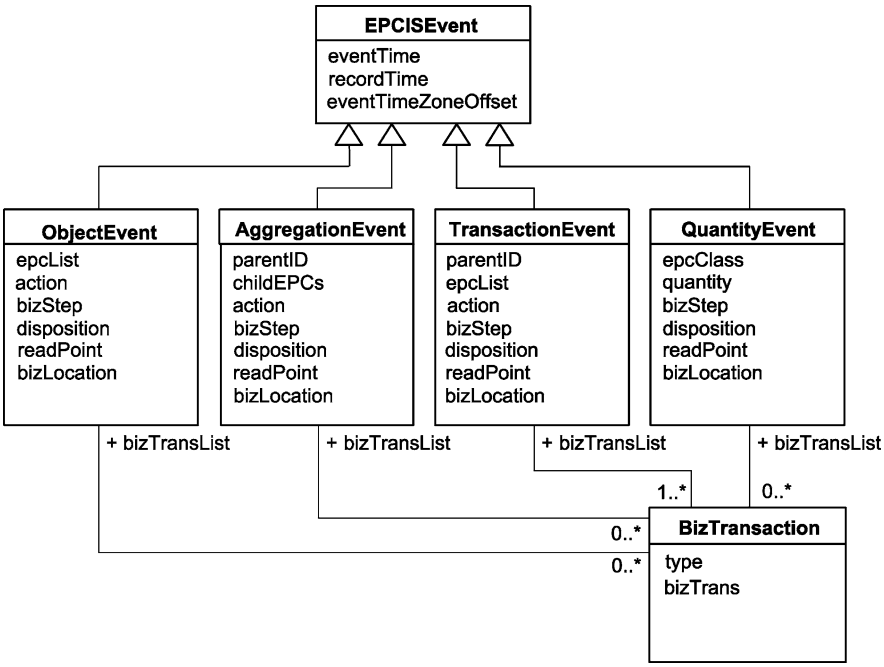


Fig. 3.3 EPCglobal's EPCIS data model

allows applications to specify and manage queries for event data using a query control interface. Querying can be done on-demand ("pull" approach) or by using the control interface to define and register standing queries that are executed periodically ("push" approach).

An EPCIS event can refer to anything happening in a supply chain that can be linked to a physical item and a discrete date. Each event makes a statement about the what, where, when, and why of a supply chain event.

- The *what* dimension is specified by a list of EPCs that identify one or several physical objects and a list of so-called business transactions that these items are involved in. A business transaction can, for instance, be a production order.
- The *when* of an event is established by two time stamps that specify the time the event happened and when it was captured.
- The *where* dimension is specified by the two variables, readPoint and bizLocation. The readPoint value is expected to be some technical ID, whereas the business location provides the corresponding context information.
- The *why* refers to a business step (bizStep) and disposition ID (disposition) that denote the state of the physical item by the time its EPC is read and its disposition after that moment.

The EPCIS data model defines four event types. An `ObjectEvent` captures information about an event that pertains to one or several physical objects identified

by EPCs. Its mandatory attributes are the event and record times inherited from `EPCISEvent`; a list of EPCs; and an action attribute that can have three values: `ADD`, `OBSERVE`, and `DELETE`. If the value of the action attribute is set to `ADD`, it means that the EPCs were associated with the physical object for the first time. `OBSERVE` means that the object has been observed, whereas `DELETE` signifies that the EPCs listed in this event were decommissioned as part of the event. All other attributes of the object represent optional context information and have to be provided by other enterprise applications before the event gets stored in the database.

An `AggregationEvent` describes events that pertain to objects that have been physically aggregated (e.g., products in a box). The action attribute uses the same semantics: `ADD` signifies that the aggregation has been observed for the first time in this event, whereas `DELETE` means that it has been decommissioned, i.e., child tags are no longer associated to a parent tag but are still in existence.

The class `QuantityEvent` represents events that take place with respect to a specified quantity of some type of objects. This event could be captured, for instance, when the inventory level needs to be reported.

Depending on the value of its action attribute, a `TransactionEvent` describes the association or disassociation of physical objects to one or more business transactions. Its structure is similar to the `ObjectEvent` class except that it has to be associated with at least one business transaction.

For more information on the EPCIS events, refer to Hribernik et al. (2007).

3.3 Business Application

Fierce competition and the resulting pressure to reduce costs while maintaining high customer satisfaction has drawn attention to possible ways to improve supply chain wide coordination. Collaboration in this context means that several independent organizations work together to achieve the common goal of supply chain wide cost reduction (Chopra and Meindl 2004; Simatupang and Sridharan 2005). Supply chain collaboration is a growing field of research; however, most collaborative efforts have so far been focusing on the demand side (VICS 1998; Waller et al. 1999): Sharing information on historical or expected demand and planning production jointly can greatly reduce common supply chain inefficiencies caused by phenomena such as the bullwhip effect (Lee et al. 1997). Although more advanced identification technology can help to increase downstream inventory accuracy (Atali et al. 2006), it has often been argued that the many benefits of standardized auto ID technologies can be obtained from the ability to track items as they are moving through the supply chain (Gaukler 2005). Interestingly, short-term coordination of supply processes using upstream information sources has received little attention in the operations community to date (Chen 2003).

In order to optimize short-term operations, decision-makers along the supply chain need to be informed about problems at upstream stages and their options for dealing with a particular problem. The short-term actions available to steer supply vary according to individual supply chain characteristics: in long- and medium-haul

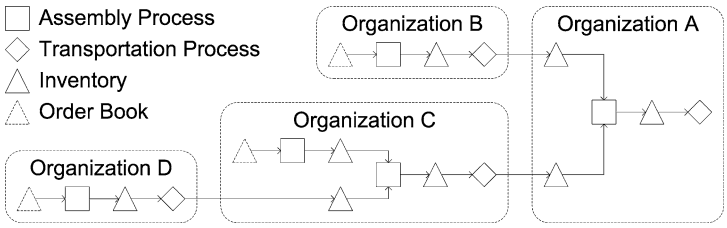


Fig. 3.4 Example of formalized assembly network

transportation there often exists the possibility to choose among different transportation modes, e.g., sea, sea/air, and air; the picking process taking place in warehouses can be accelerated if needed, for instance, by skipping certain quality assurance processes; or capacity can be added to production processes, e.g., by increasing machine throughput or by extending shifts. Information on the available short-term control options is usually only valid for a very short period of time; thus, any event management system designed to support operational supply chain management has to include a component capable of transmitting and offering up-to-date control options.

To be able to analyse the problem of short-time management of assembly networks in a structured manner, we introduce the semantics of a simple formalization of such networks in the following. According to Chopra and Meindl (2004), the four drivers of supply chain management are facilities, inventories, transportation, and information. The way that these drivers are applied determines the performance and operational cost of a supply chain. Each of the four drivers will be reflected in our formalization. According to our model, an assembly network consists of one or more supply chain organizations. A supply chain organization in turn consists of an arbitrary number of internal nodes which can either be an assembly process node, an inventory node, or a transportation node. Internal nodes are connected by edges indicating the flow of material. Assembly and transportation processes always need to be decoupled by an inventory node. Furthermore, one inventory node always refers to one particular item type. The upstream end of the formal assembly network is marked by order book nodes. Each order book holds the production orders for a subsequent assembly node.

Figure 3.4 shows an example assembly network consisting of four supply chain organizations forming a three-tiered assembly network. The network conforms to the rules stated above. We will use this example throughout the section to illustrate the working of our event-based architecture.

The information required to optimize the coordination of an assembly network basically consists of schedules, i.e., events that are expected to take place at certain dates, the events actually taking place as material moves downstream, and the relevant control options. The purpose of the system architecture proposed in Sect. 3.4 is to provide all nodes in the network with the technical means to share the required information in a decentralized way. A coordination mechanism that determines the optimal control option in case the scheduled events do not match the expected events within a certain range of tolerance is deliberately not part of this work.

3.4 Decentralized EPCIS-Based SCEM

In this section we propose an extension to the EPCIS specification and describe a process that utilizes this extension for providing participants of a supply chain with additional planning capabilities and enabling them to timely detect delays and promptly react to them. The proposed extension affects data, protocol, and application layers of the EPCIS specification. Below we provide a detailed description of these modifications.

3.4.1 Data Layer

Using a common data format like the one specified by EPCglobal to store EPC-related data is definitely valuable in providing interoperability between applications used in one organization and for the interchange of event data between organizations. However, the business application described in Sect. 3.3 requires context data in the shape of expected events; therefore, we extended the EPCIS event data framework by the class ExpectedEvent (see Fig. 3.5).

Interorganizational sharing of event data can also be done using an ONS or a Discovery Service; however, this results in a centralized query infrastructure in the

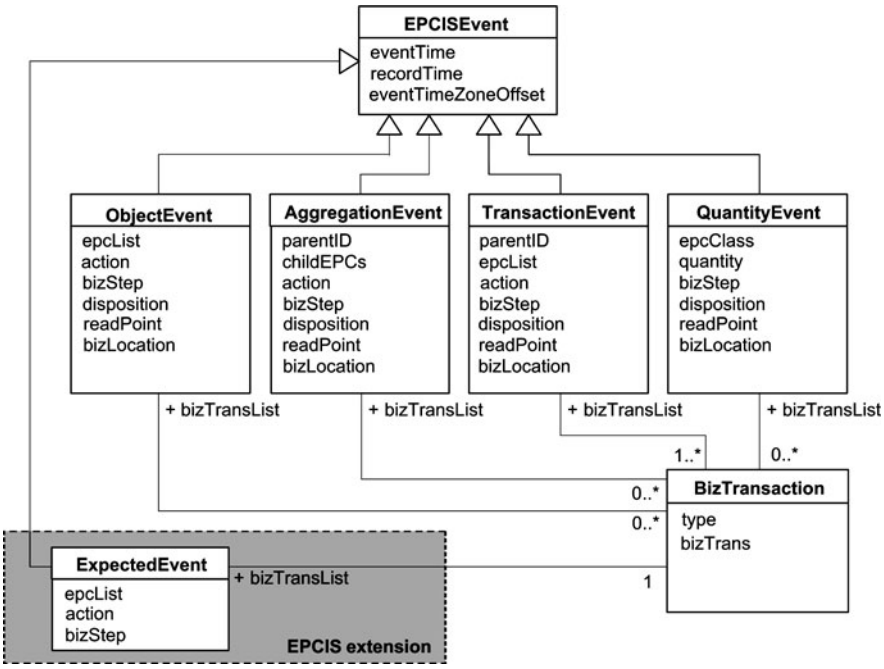


Fig. 3.5 Extended EPCIS data model

hands of EPCglobal with the two mentioned services representing single points of failure. Furthermore, the context data required to make sense of the event data would have to be shared via an additional, unstandardized communication channel. In our proposal we strive to specify an architecture that takes advantage of existing bilateral business relationships in the supply chain.

3.4.2 Protocol Layer

The entities communicating on the protocol layer are the nodes of the assembly network. Within our architecture, these nodes represent communication hubs and controllers at the same time. Each node in the assembly network maintains a list of predecessors and a successor node for each type of product. Upstream messages are sent to some subset of predecessor nodes while downstream messages are sent to the successor node. Different product types have different bills of material, i.e. nodes would maintain at most one predecessor and successor list for each product type.

The communication taking place to coordinate the assembly of products is separated into six phases. Figure 3.6 presents an overview of the entire protocol. During phase one, lead times are quoted recursively. Each node implementing the protocol's communication primitives can query its upstream assembly network in order to find out if a certain delivery date can be met. Answering the query implies searching the assembly tree of a particular product type for the maximum lead time path. The answer consists of the date by which the order has to be issued at the root node in order to meet the requested delivery date. There are two message formats defined for this communication phase: an upstream message, called leadtimeRequest, containing the attributes productType and endDate, and a downstream message, called leadtimeQuote, containing the attribute startDate. Phase 1 is given as pseudocode below:

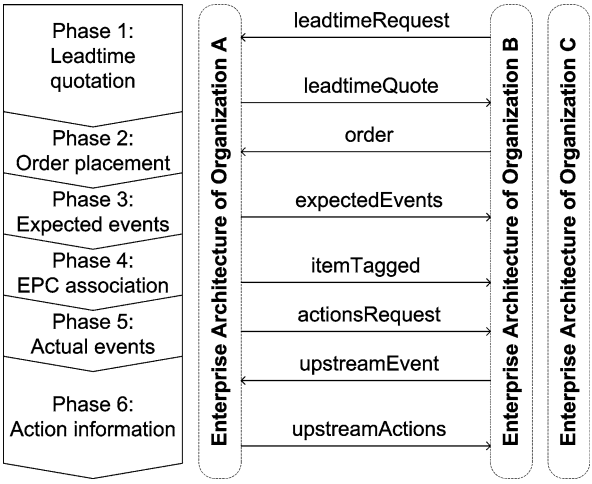


Fig. 3.6 Protocol layer for EPCIS-based SCEM

- Upon reception of leadtimeRequest(productType:productTypeID, endDate:Date) by node i:
  - If node i is of type orderbook:
    - Set startDate to the earliest startDate incremented by node i's expected duration
    - Send leadtimeQuote(productType:productTypeID, startDate:Date) to involved successor node
  - Otherwise:
    - Send leadtimeRequest(productType, endDate) to all corresponding predecessor nodes
- Upon reception of leadtimeQuote(productType:productTypeID, startDate:Date) by node i from all involved predecessor nodes:
  - Set startDate to the maximum startDate quoted by the predecessors incremented by i's expected duration
  - Send leadtimeQuote(productType:productTypeID, startDate:Date) to the involved successor node

In our example, if the root node of organization A initiates the request, it will eventually end up with the expected lead time of the entire assembly process. From the value of startDate it can infer whether the order can be filled before the requested delivery date or not. If the quoted startDate has already passed, another query using a later delivery date can be initiated.

Order propagation constitutes the second communication phase. In our example, upon reception of a customer order, organization A initializes an upward information diffusion process of order data: A's root node sends an order message to its predecessors indicating that an order has been issued. The message contains a unique order ID and the scheduled date of delivery. Each order ID is represented by a BizTransaction object. The predecessor nodes propagate the order ID and the scheduled delivery date decremented by their respective expected process durations. The propagation process terminates when an order book node is reached; thereafter, the order is stored in the order book until the transmitted date coincides with the actual time. If this happens, the assembly process represented by the successor node is triggered. Phase 2 is given as pseudocode below:

- Upon reception of order(orderID:BizTransaction, deliveryDate:Date) by node i from successor:
  - Set deliveryDate to the deliveryDate sent by the successor decremented by i's duration
  - Send order(orderID, deliveryDate) to all involved predecessor nodes

The third communication phase consists of messages containing ExpectedEvent objects which are sent downstream. The expectedEvents messages used in this phase serve to let downstream nodes know when certain items are scheduled to enter and leave each node. The ExpectedEvent class, which is used to store expected events, represents an extension of the EPCglobal EPCIS standard. We embedded the event type ExpectedEvent as child of EPCISEvent (see Fig. 3.5). According to EPCglobal,

adding a new event type implies updating the EPCIS standard specification. In our case the semantics of the EPCISEvent class would have to be adapted to include the possibility of events that have not yet taken place. Upon reception of an expectedEvents message concerning a particular order from all involved predecessors, a node remembers which events are scheduled to take place in the future by storing them in its local event repository or in the volatile storage of an SCEM application. It then creates the events it expects to happen at its own entry and exit points. As indicated by Fig. 3.5, an expected event requires the attributes epcList, action, and BizStep. By the time an ExpectedEvent object is created, there are no EPCs stored as values of its epcList attribute. If the object is created in response to an order, the action attribute is set to ADD. In case expected events need to be withdrawn, for instance, because the corresponding order was canceled, the action attribute is set to DELETE. The BizStep attribute is needed as a key to later match the expected with the actual events and is either set to the BizStepID of the entry or the exit point of the node. Newly created ExpectedEvent objects are combined with the received objects into a new set and sent downstream. Phase 3 is given as pseudocode below:

- Upon reception of expectedEvents(expectedEventSet:Set[Expected-Event]) pertaining to a particular BizTransaction by node i from all involved predecessors:
  - Capture all ExpectedEvent objects contained in all expectedEventSets
  - Merge all expectedEventSets to obtain mergedExpectedEventSet
  - Create own ExpectedEvent objects and add them to mergedExpectedEventSet
  - Send expectedEvents(mergedExpectedEventSet:Set[Expected-Event]) to the involved successor node

Phase 4 serves to complete the expected events created in phase 3 by the EPCs. This information is needed to identify pairs of expected and actual events which have to be compared in order to detect delays. We assume that EPCs are allocated at about the same time that physical objects are associated with an EPC. We believe that this is a reasonable assumption considering practical constraints such as RFID printers which store fixed EPCs on passive tags. When a physical object gets associated with an EPC at some node, this node sends an itemTagged message to its successor. Each message of this type contains an EPC and the keys required to map the allocated or removed EPC to event entries at downstream nodes. Furthermore, it contains the type of action to be triggered by the message, i.e., either association or disassociation of EPC and expected event. When all stored ExpectedEvent objects have been enabled by adding one or several EPCs, each node possesses the information it needs to identify delays as upstream events of any type. Phase 4 is given as pseudocode below:

- Upon reception of itemTagged(epc:EPC, orderID:BizTransID, nodeStep: BizStepID, action:ActionID) by node *i* from a predecessor:
  - If action is ADD:
    - Add EPC to all previously captured ExpectedEvents with the corresponding BizTransID and BizStepID
  - If action is DELETE:

  - Remove EPC from all previously captured ExpectedEvents with the corresponding BizTransID and BizStepID
- Send itemTagged(epc:EPC, orderID:BizTransID, nodeStep:BizStepID, action:ActionID) to involved successor node

In phase 5, messages of type upstreamEvent are being sent downstream to spread the news on actual events taking place upstream. Each of them carries an EPCISEvent object including the attached BizTransaction object which refers to the order. It would be straightforward to only use the generated ObjectEvent objects in the protocol since they are created at all process steps. Phase 5 is given in pseudocode below:

- Upon capturing of event:EPCISEvent at node i:
  - Send upstreamEvent(event:EPCISEvent) to the involved successor node
- Upon reception of upstreamEvent(event:EPCISEvent) by node i from a predecessor:
  - Capture event
  - Forward upstreamEvent(event:EPCISEvent) to the involved successor node

The final phase of the communication protocol allows each node to collect up-to-date action alternatives to make up for a particular delay. By comparing the dates of expected and actual events that have been captured during the previous communication phases, a node can identify upstream delays; however, in order to exert control, the node requires information about which actions can currently be taken to influence the processing of a particular order. The path of nodes between the node that caused the delay and the node that identified the delay, we refer to as the action path of a delay. Our protocol provides the opportunity to query the upstream network for these action paths. Any node can initiate such a query by sending a message of type actionsRequest to all its predecessors. This message contains three attributes: the EPCs that the delayed event refers to, the orderID of the delayed order, and the BizStepID of the processing step where the delay occurred. When an upstream node receives a message of type actionsRequest, it first checks whether it has stored an ExpectedEvent containing the EPCs in the message. If this is the case, it compares the BizStepID with the one of its exit points. If the two BizStepIDs are not equal, it forwards the message to all of its predecessors that are involved in the assembly process; otherwise, the node which has caused the delay has been reached. This node then creates a message of the type upstreamActions containing information on all possible actions that can be taken to speed up order processing at its site and forwards the message to its successor. If the successor is not the original requester, it adds its own ways to deal with delays concerning this order and sends the message to its own successor. This way the original requester ends up with a list of all up-to-date opportunities along the action path to speed up a particular order. Phase 6 is given in pseudocode below:

- Upon reception of actionsRequest(EPCs:Set[EPC], orderID:BizTransaction, delayedStepID:BizStepID) by node i from successor:
  - If an ExpectedEvent containing EPCs exists:

  - If delayedStepID equals exitStepID:
    - Retrieve available speedup actions for EPCs and orderID
    - Send message upstreamActions(actions) to involved successor
  - Otherwise:
    - Send message actionsRequest(EPCs:Set[EPC], orderID:BizTransaction, delayedStepID:BizStepID) to all involved predecessors

- Upon reception of upstreamActions(EPCs:Set[EPC], orderID:BizTransaction, actions:Set[Action]) by node i from a predecessor:
  - If node i is the original requester:
    - Evaluate and trigger actions
  - Otherwise:
    - Retrieve available speedup actions corresponding with EPCs and orderID
    - Append these speedup actions to actions
    - Send message upstreamActions(EPCs:Set[EPC], orderID:BizTransaction, actions:Set[Actions]) to the involved successor

3.4.3 Application Layer

Having presented the protocol layer of our architecture in the previous section, we now turn to its application layer. The application layer consists of all enterprise systems that use the primitives of the protocol described in Sect. 3.4.2.

Capacity in the shape of production slots, warehouse space, or transportation capacity is usually managed by a corresponding information system which forms part of an Enterprise Resource Planning (ERP) solution. The quotation of process durations in phases 1 and 2 of the communication protocol described in Sect. 3.4.2 thus depends on the input from those systems. The data that has to be provisioned to the protocol includes the quotable process start dates, the identifiers of entry and exit points of nodes in the assembly network, and the available speedup actions. Customer facing systems such as order management provide other inputs required for the working of the protocol. These inputs include the requested delivery date for an order, the type of product to be assembled, and the allocated order IDs. Order management forms part of most standard ERP solutions.

EPCs are allocated by the EPC management of an organization. When a new EPC is created and attached to a physical object, this information needs to be published on the protocol layer.

The application layer component of the EPCIS-based decision support architecture required at each node consists of two components: A local EPCIS implementation and an SCEM application interfacing with users. Expected and actual events are stored in local EPCIS repositories that need to be accessed by the SCEM application to identify delays. The SCEM application also needs to have direct access to the protocol layer in order to retrieve action paths.

Figure 3.7 depicts the general layout of the proposed architecture. The three components Supply Chain Event Management, EPC Management, and EPC Information

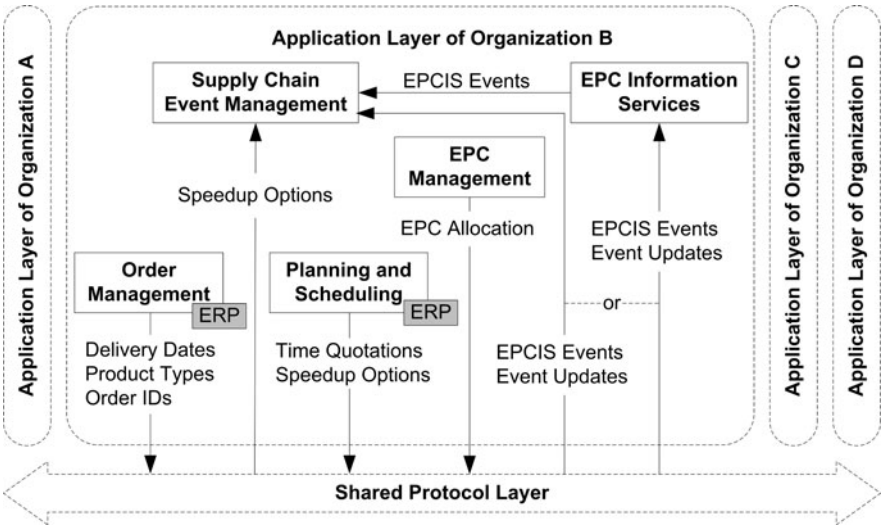


Fig. 3.7 Two-layered EPCIS-based architecture for SCEM

Services have to be added to the existing ERP solution in order to let an organization take advantage of the data being transmitted in the protocol layer.

3.5 Quantitative Comparison of Two Architecture Approaches

In the previous section we described a communication protocol based on the EPCIS data format that allows for distributing scheduling and monitoring data within an assembly network. Based on the data transmitted on this protocol layer, the supply chain stake holders can estimate lead times, detect delays, and obtain complete information about speedup options. In our proposal we assumed that the communication hubs exchange event data according to established bilateral relationships determined by the structure of the assembly network. In this section we aim at providing an objective comparison of this approach with the approach implied by the current EPCIS specification, in particular the design of the EPCIS Discovery Service used for identifying EPCIS repositories along the supply chain that contain data related to a given EPC. In the following, we briefly describe the methods applied to evaluate and compare both architectural designs, present the evaluation results, and discuss their implications.

3.5.1 EPCIS-Based Event Sharing Using Event Pull

EPCglobal proposes a centralized query infrastructure which can be used to retrieve all events relating to a particular EPC from all accessible EPCISs worldwide. The

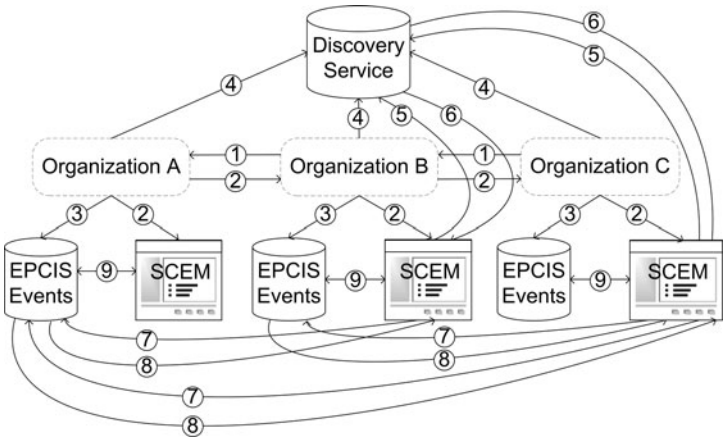


Fig. 3.8 Event Pull in a three-tiered supply chain

retrieval process has two steps: first, the EPCIS Discovery Service (DS) is queried for a set of references to all EPCISs which have stored events involving a particular EPC. Upon receiving this set, the query interfaces of all EPCIS in the set can be directly queried for particular types of events, i.e., the range of events searched for can be restricted to the information of interest. This architecture is well suited for situations when there is no ex ante knowledge about the applications which will use it. In principle, it allows for the retrieval of EPCIS events based on arbitrary search criteria which have previously been stored in any EPCIS; therefore, it can also be used to realize SCEM.

Figure 3.8 describes how the event-sharing mechanism works in the context of SCEM if the architecture approach of EPCglobal is followed. The concrete steps are as follows:

1. The last organization (C) in the supply chain places an order with its supplier (B), which in turn places an order with its own supplier (A).
2. The first organization in the chain (A) schedules activities, translates the schedule into expected events, stores these events in its SCEM application, and sends the set of expected events downstream. The next organization in the supply chain schedules its own activities relating to the order, adds the corresponding expected events to the set, saves all events in its SCEM application, and sends the extended set downstream and so forth.
3. Actual events are continuously captured by the EPCIS.
4. Each time an actual event is captured, the organization publishes the event's availability to the EPCIS DS: in this case a key-value pair of EPC and EPCIS reference.
5. If the SCEM application wants to request the status of an EPC, it has to query the EPCIS DS to receive the address of the relevant EPCIS repositories; alternatively, a so-called standing query can be saved so that a foreign EPCIS does not need to be polled continuously.

6. The addresses of the EPCIS repositories which contain event data related to the EPCs of the order are sent to the SCEM application.
7. The SCEM applications separately and directly query each EPCIS repository which contains relevant events.
8. The EPCIS repositories send the event data requested by the downstream SCEM applications.
9. The SCEM applications constantly compare scheduled with actual events.

3.5.2 EPCIS-Based SCEM Using Event Push (Our Proposal)

Contrary to the approach described in the previous section, events can be exchanged according to the bilateral relationships in a supply chain, e.g., between a manufacturer and its suppliers. Instead of replying to concrete requests from downstream organizations, upstream organizations can simply push all events relevant for SCEM to them. These events can be forwarded downstream without a previous request because upstream organizations know which events are relevant from previous interaction, e.g., the sharing of schedules. In this alternative architecture, the supply chain also serves as a type of communication network at the same time; therefore, data only needs to be exchanged by parties which are already involved in a business relationship.

Figure 3.9 describes the Event Push. Steps 1 and 2 are the same as Event Pull above. The following steps are:

3. Actual events are continuously captured and immediately sent to the adjacent downstream supply chain organization, which does the same and so forth.
4. The SCEM applications constantly compare scheduled with actual events.

3.5.3 Evaluation

In order to allow for a rigorous comparison of the two architectures, we outline how they work in detail in the following sections. In spite of a number of qualitative criteria which may also have an influence on which of the proposed architectures will be preferred in practice, we will focus on quantitative

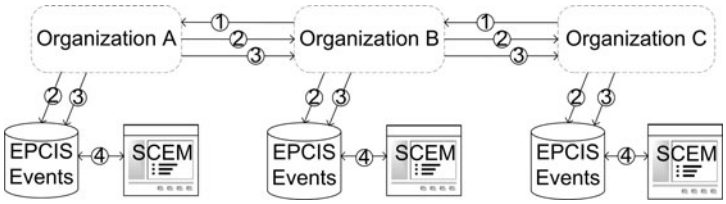


Fig. 3.9 Event Push in a three-tiered supply chain

measures for evaluating and comparing the two system architectures presented in Sect. 3.5. The performance criteria we use refer to three of the most frequently mentioned performance characteristics of information systems (Bocij et al. 2005; Garcia et al. 2006): efficient use of network capacity, efficient use of storage, and system reliability. These performance criteria were operationalized by quantitative performance metrics based on the number of data objects stored along the supply chain and the number of messages exchanged between supply chain stages.

The performance metrics depend on several parameters which characterize the structure of and the flow of material in the supply chain; thus, the evaluation of the EPCIS-based SCEM architecture depends on performance metrics which inherently reflect the particularities of the supply chain context. Parameters and performance metrics will be formally defined in the following sections.

3.5.3.1 Parameters

A supply chain is composed of at least two organizations, tiers, sites, or stages which work together in order to provide one product to the end customer. The number of tiers in each supply chain is denoted by $l \in N^+ \setminus \{1\}$. The number of supply chains which are monitored by the SCEM application is denoted by $d \in N^+$. Note that for the sake of simplicity, we do not consider intermeshed supply chains; intermeshed supply chains come into existence if at least one company takes part in two different supply chains and the organizations in these supply chains are not the same. The last parameter which is considered in our analysis is the number of tagged components or products which move through each supply chain during a fixed period of time. This parameter is denoted by $p \in N$.

3.5.3.2 Efficient Use of Network Capacity

The efficient use of available network capacity is measured in terms of the absolute number of messages exchanged during a fixed period of time. A message in this case is defined as a temporarily enclosed and distinct exchange of data between the information systems of different organizations. Since the two system architectures to be compared do not differ regarding the way in which order data and schedules (or expected events) are forwarded along the supply chain, these steps are not included in the number of messages exchanged.

In the Event Push approach, the actual events which are forwarded along the supply chain are the only remaining messages. The amount of messages exchanged grows multiplicatively with the depth of supply chains; events are managed separately for different supply chains.

Consider, for example, a supply chain involving two organizations A and B; one message is sent from organization A to organization B when an event related to one product has been captured by A. If the supply chain is extended by one organizational tier (Organization C), not only the captured events of B, but also those

captured by A are sent from B to C (B serves as a communication hub in this case). The number of exchanged messages in Event Push can be calculated in the following way:

$$M_{\text{push}} = d \cdot \sum_{k=1}^{l-1} k \cdot p = \frac{1}{2} \cdot d \cdot p \cdot \left(l^2 - l\right). \tag{3.1}$$

In the Event Pull approach, captured events are not forwarded to subsequent organizations in the supply chain but rather pulled from upstream organizations on demand. Again, the amount of exchanged messages grows multiplicatively with the number of supply chains the organizations are involved in. For each EPC that is read by an organization, the corresponding key-value pair has to be published via the EPCIS DS (step 4). In order to compare an expected event with the corresponding actual event, an SCEM system has to query the EPCIS DS for the reference to an EPCIS repository (steps 5 and 6). For each received reference, SCEM systems have to query the EPCIS repository for the corresponding EPCIS event (steps 7 and 8); therefore, the number of exchanged messages in the Event Pull approach is

$$M_{\text{pull}} = d \cdot \left[l \cdot p + 4 \cdot \sum_{k=1}^{l-1} k \cdot p \right] = d \cdot p \cdot \left(2 \cdot l^2 - l\right). \tag{3.2}$$

Event Push dominates Event Pull in terms of the number of exchanged messages. The factor with which the push approach performs better can be calculated using the following formula:

$$\delta_M = 4 + \frac{2}{l-1}. \tag{3.3}$$

As (3.3) indicates, the number of supply chains d and products p do not play a role when comparing the performance of the two proposed architectures with respect to their use of network capacity: the performance advantage of Event Push only depends on the length l of the supply chains. The number of exchanged messages produced by Event Pull is six times higher than the one produced by Event Push for supply chains with two participants ($l = 2$), 4.5 times higher for $l = 5$, 4.2 times for $l = 10$, and approaches 4 times higher for high values of l.

3.5.3.3 Efficient Use of Storage Capacity

The efficient use of storage capacity by the two architectural approaches is measured in terms of the number of stored data objects which refer to the flow of goods. We initially compare the number of events saved in the EPCIS repositories at the different supply chain participants.

In Event Push, each supply chain participant stores its expected and actual events at its own and all subsequent sites. The number of supply chains affects this number

multiplicatively. The number of saved EPCIS events can be calculated using the following formula:

$$O_{\text{push}} = 2 \cdot d \cdot \sum_{k=1}^{l} k \cdot p = d \cdot p \cdot \left(l^2 + l\right). \tag{3.4}$$

In Event Pull as proposed by EPCglobal, schedules would not be stored in the form of events within the EPCIS repositories but would be directly exchanged by the SCEM applications; therefore, the number of stored EPCIS events can be calculated according to formula (3.5):

$$O_{\text{pull}} = d \cdot p \cdot l. \tag{3.5}$$

However, Event Pull requires the storage of other data objects. Both the key-value pairs used as references in the EPCIS DS and the expected events stored separately by the SCEM applications have to be taken into account. Thus, a fair basis for comparison regarding the number of stored data objects in the pull approach is given by formula (3.6):

$$\bar{O}_{\text{pull}} = 2 \cdot d \cdot p \cdot l + d \cdot p \cdot \sum_{k=1}^{l} k = d \cdot p \cdot \left(\frac{1}{2}l^2 + \frac{5}{2}l\right). \tag{3.6}$$

No approach formalized in functions (3.4), (3.5), and (3.6) is dominated with respect to the number of stored data objects. The relative advantage of Event Push over Event Pull (or vice versa) expressed by formula (3.7) is independent of the number of supply chains and the number of products moving through each of them.

$$\delta_O = 2 - \frac{8}{l+5}. \tag{3.7}$$

The number of stored objects is 1.2 times higher if Event Pull is used for two supply chain participants, equal for three participants, 1.2 times smaller for five participants, 1.5 times smaller for ten participants, and approaches 2 times smaller for high values of l.

3.5.3.4 Reliability

The number of data objects stored at each supply chain participant should not be much above the average number in order to minimize bottlenecks and maximize reliability. We operationalize this performance criterion by measuring how dispersed the required data objects are stored in the supply chain. A standard measure of statistical dispersion is the Gini coefficient G. The value of G ranges from 0 to 1; the nearer it is to 1, the greater the dispersion. Since reliability is expected to be greater if data objects are distributed more equally among the databases along the supply chain, a lower Gini coefficient of the number of stored objects indicates higher reliability.

Table 3.1 Comparison results

(a)

l	d	G_{push}	G_{pull}	$\frac{G_{\text{push}}-G_{\text{pull}}}{G_{\text{pull}}}$
2	1	0.333	0.381	12.5%
2	2	0.458	0.471	2.8%
2	3	0.500	0.531	5.8%
3	1	0.444	0.438	−1.6%
3	2	0.528	0.530	0.4%
3	3	0.556	0.575	3.4%
4	1	0.500	0.478	−4.7%
4	2	0.563	0.562	−0.1%
4	3	0.583	0.598	2.5%
[···]				
10	1	0.600	0.577	−4.0%
10	2	0.625	0.624	−0.2%
10	3	0.633	0.641	1.2%

(b)

l	d	G_{push}	G_{pull}	$\frac{G_{\text{push}}-G_{\text{pull}}}{G_{\text{pull}}}$
2	1	0.333	0.381	12.5%
2	100	0.581	0.673	13.7%
2	10000	0.583	0.679	14.0%
3	1	0.444	0.438	−1.6%
3	100	0.609	0.677	10.0%
3	10000	0.611	0.681	10.2%
4	1	0.500	0.478	−4.7%
4	100	0.624	0.678	8.0%
4	10000	0.625	0.681	8.2%
[···]				
10	1	0.600	0.577	−4.0%
10	100	0.650	0.676	3.9%
10	10000	0.650	0.677	3.9%

We do not compare the Gini coefficients of data dispersion for the two system architectures formally since the derivation of a mathematical expression is highly complex if feasible at all; instead, we base our analysis on a numerical comparison. Table 3.1(a) shows the relevant results of the numerical calculations and provides the relative performance differences between both architecture approaches. The performance metrics are invariant with respect to the number of products p but depend on the depth d of the supply chains.

When comparing the architecture approaches based on our reliability metric, several impacts of the parameters l and d can be observed. The longer the supply chain becomes, the smaller the advantage of Event Push compared to Event Pull. The more supply chains there are, the greater the advantage of Event Push becomes. Table 3.1(a) shows that if the number of supply chains is very low, the push approach can have a higher Gini coefficient; however, as Table 3.1(b) shows, this disadvantage of Event Push only persists up to parameter configuration with $d = 2$, i.e., it should be negligible in realistic settings.

3.5.4 Results

Supply chain wide visibility of the flow of goods is a precondition for supply chain event management. We have compared two possible system architectures that enable the sharing of standardized supply chain event data with respect to a number of quantifiable criteria. According to our evaluation, none of the approaches can be preferred without further consideration.

Table 3.2 Relative advantage of event push over event pull

Length of supply chain l	Network capacity	Storage capacity	Reliability
2	83.3%	14.3%	14.0%
3	80.0%	0.0%	10.2%
4	78.6%	−11.1%	8.2%

The parameters we used to evaluate and compare the two architectures are realistic variable values for length, depth, and number of products. Iyengar (2005) calculated the average length of supply chains using the US Benchmark Input–Output tables published by the Bureau of Economic Analysis. Based on data from more than 1 million supply chains, he found that in 1997 the average US supply chain had a length between 3.4 and 4.1 depending on the industry. Length was defined as the number of echelons of the supply chain. On the basis of these figures, it seems realistic to consider supply chains consisting of two to four participants.

Estimating a realistic number of supply chains, which would benefit from SCEM applications, and the number of products flowing through these supply chains is considerably more difficult but can be expected to be very high. Kürschner et al. (2008) state that the EPCIS Discovery Services will have to be able to handle queries from millions of clients. Against this background, our estimation of 10,000 supply chains, which are monitored using an SCEM application, should be realistic.

Table 3.2 summarizes the relative advantage of Event Push compared to Event Pull with respect to the quantitative metrics defined in Sect. 3.5.3.1 and based on realistic parameter values. In spite of the typical trade-off between usage of data storage and network bandwidth, Event Push appears to be the preferable architectural choice for short supply chains: up to a supply chain length of three echelons, the push approach dominates the pull approach according to our criteria.

3.6 Discussion

We have presented a business application and a corresponding information system architecture that provide the basis for the short-term coordination of a multiorganizational assembly network. The proposed system architecture was chosen for a number of reasons, each of which can be attributed to the requirements of short-term decision support in dynamic multiorganizational business environments, in particular system interoperability and the interorganizational management of EPC context data.

We have chosen to address the informational needs of our business application in order to derive concrete requirements. The concept we describe comes near to what is known as SCEM. SCEM has found general approval in practice since it addresses a number of pressing problems in today's competitive environment. To the best of our knowledge, this work represents the first attempt to suggest possible ways to realize SCEM applications based on the EPCglobal specifications while

taking their specific requirements regarding interoperability and systems integration in multiorganizational environments into account.

From an operational point of view, an obvious shortcoming of the proposed architecture is that it does not address dynamic scheduling. Although it allows for order cancellation, the schedule of other orders encoded in the form of ExpectedEvent objects throughout the network cannot be changed in response to such an event. Certainly the protocol layer could be extended in order to deal with dynamic scheduling, but it remains to be seen if such an extension is feasible in practical circumstances. Another limitation of the architecture results from its bilateral character. Messages are forwarded along the supply chain, i.e., if an organization in the middle of the supply chain does not implement the protocol, our approach will not work. This problem could be solved by a third party willing to act as a trusted communication intermediary.

The proposed architecture supports interoperability in two ways: first, due to its two-layered design, there is no need to standardize any components on the application layer which facilitates the development and integration of the EPC/SCEM components; second, one common way to describe event data and its context based on the EPCglobal event data specification is used both for intra- and interorganizational communication.

In our application, up-to-date context data required by downstream nodes and organizations gets distributed without former request as soon as it becomes available. This approach relieves the burden of downstream organizations from the need to maintain a comprehensive up-to-date internal process view of other organizations. Furthermore, ex ante knowledge of the organizational structure of the assembly network is not required, which represents a crucial advantage in today's dynamic and complex supply chains. Synchronization of data and context is assured by design since data and context are sent via the same communication channel.

The second research question has been whether the current proposal for the distributed system architecture of the Internet of Things is suitable for SCEM applications. Based on three quantitative criteria, we come to the conclusion that an alternative approach based on the idea of pushing EPCIS events downstream could be the preferable choice. We have also mentioned some qualitative advantages of the latter architecture, such as taking advantage of existing business relationships in the supply chain and not requiring a central authority for data management and authentication which speak for Event Push.

3.7 Conclusions and Future Work

Although our quantitative measures are coarse and based on simplistic assumptions, they provide an objective means for an initial comparison of Event Pull and Push to realize inter-organizational SCEM.

Further research on the topic is definitely warranted: in order to make an informed decision, the relative importance of different performance criteria and metrics will have to be determined (e.g., based on the available network and data storage

capacity and on the variable costs of storing and transmitting data). Furthermore, additional criteria and metrics should be defined to obtain a more detailed picture of possible cost-benefit trade-offs. For instance, operational properties such as latency and throughput could be measured and compared using supply chain simulations as soon as implementations of the proposed architectures are available. Finally, economic translations of the somewhat technical performance measures used in this work have to be defined in order to enable a sound investment decision by adopters of EPCIS-based SCEM.

We see a number of promising areas for further research on the proposed architecture. First of all, the architectural design needs to undergo further validation. Secondly, it needs to be extended to cope with dynamic rescheduling. The business logic of the actual decision support system, i.e., the development of algorithms used to optimize courses of action based on action path data, are a promising research direction. Still another issue that needs to be dealt with is authentication and security. The communication taking place on the protocol layer needs to be secured against malicious behavior, e.g., by using dedicated public key infrastructures.

References

Atali, A., Lee, H., Özer, Ö.: If the Inventory Manager Knew: Value of Visibility and RFID under Imperfect Inventory Information. In: Manufacturing and Service Operations Management Conference, Evanston, IL (2006)

Bocij, P., Chaffey, D., Greasley, A., Hickie, S.: Business Information Systems: Technology, Development & Management for the E-Business, 3rd edn. Prentice Hall, Upper Saddle River (2005)

Chen, F.: Information Sharing and Supply Chain Coordination. In: T. de Kok, S. Graves (eds.) Supply Chain Management: Design, Coordination and Operation, pp. 341–421. Elsevier, Amsterdam (2003)

Chopra, S., Meindl, P.: Supply Chain Management. Strategy, Planning, and Operations. Prentice Hall, Upper Saddle River (2004)

Chow, H.K.H., Choy, K.L., Lee, W.B., Chan, F.T.S.: Integration of Web-based and RFID Technology in Visualizing Logistics Operations—A Case Study. Supply Chain Management: An International Journal **12**(3), 221–234. (2007)

EPCglobal: EPC Information Services (EPCIS), Final Version 1.0.1 (2007)

EPCglobal: Object Naming Service Standard Version 1.0.1. http://www.epcglobalinc.org (2008a)

EPCglobal: Tag Data Standard Version 1.4 (2008b)

EPCglobal: The Application Level Events (ALE) Specification. Version 1.1 Part I: Core Specification (2008c)

EPCglobal: The EPCglobal Architecture Framework, Version 1.3 (2009)

Floerkemeier, C., Roduner, C., Lampe, M.: RFID Application Development with the Accada Middleware Platform. IEEE Systems Journal **1**(2), 82–94 (2007)

Garcia, J.D., Carretero, J., Garcia, F., and Calderon, J.F., Singh, D.E.: A Quantitative Justification to Partial Replication of Web Contents. In: Computational Science and Its Applications, pp. 1136–1145. Springer, Berlin (2006)

Gaukler, G.M.: RFID in Supply Chain Management. Ph.D. thesis, Stanford University (2005)

Hribernik, K.A., Schnatmeyer, M., Plettner, A., Thoben, K.D.: Application of the Electronic Product Code EPC to the Product Lifecycle of Electronic Products. In: EU RFID Forum 2007, Brussels, Belgium (2007)

Iyengar, D.: Effect of Transaction Cost and Coordination Mechanisms on the Length of the Supply Chain. Ph.D. thesis, University of Maryland (2005)

Kürschner, C., Condea, C., Kasten, O., Thiesse, F.: Discovery Service Design in the EPCglobal Network—Towards Full Supply Chain Visibility. In: Proceedings of the Internet of Things 2008, Zürich, pp. 19–34 (2008)

Lee, H.L., Padmanabhan, V., Whang, S.: Information Distortion in a Supply Chain: The 'Bullwhip Effect'. Management Science **43**(4), 546–558 (1997)

Michael, K., McCathie, L.: The Pros and Cons of RFID in Supply Chain Management. In: International Conference on Mobile Business, Sydney, pp. 623–629. Sydney, Australia (2005)

Niederman, F., Mathieu, R.G., Morley, R., Kwon, I.W.: Examining RFID Applications in Supply Chain Management. Communications of the ACM **50**(7), 92–101 (2007)

Otto, A.: Supply Chain Event Management: Three Perspectives. International Journal of Logistics Management **14**(2), 1–13 (2003)

Simatupang, T.M., Sridharan, R.: An Integrative Framework for Supply Chain Collaboration. The International Journal of Logistics Management **16**(2), 257–274 (2005)

Trappey, A.J.C., Lu, T.H., Fu, L.D.: Development of an Intelligent Agent System for Collaborative Mold Production with RFID Technology. Robotics and Computer-Integrated Manufacturing **25**(1), 42–56 (2009)

VICS. V.I.C.S.A.: Collaborative Planning, Forecasting and Replenishment. http://www.cpfr.org (1998)

Waller, M., Johnson, E.M., Davis, T.: Vendor Managed Inventory in the Retail Supply Chain. Journal of Business Logistics **20**(1), 183–203 (1999)

Chapter 4
Cost-Benefit Analysis to Hedge with Third-Party Producers in Demand-Driven Production

Omar Hussain and Tharam Dillon

Summary One of the characteristics of Demand-Driven Production is that goods should be manufactured and delivered to customers within the specified period of time. Manufacturers achieve this by utilizing various efficient production planning, scheduling tools and techniques. But situations may arise where the manufacturer, despite such techniques, may not be able to meet the required demand. So strategies need to be developed by which situations like these are countered and the financial loss from them alleviated. One such strategy is to hedge the production of goods from third-party producers. But before doing so, the manufacturer has to carry out a cost-benefit analysis that will determine the feasibility and viability of considering this option. In this chapter, we propose a methodology by which the manufacturer does the cost-benefit analysis and then makes an informed decision about whether to hedge with third-party producers.

4.1 Introduction

In Demand-Driven Production, the basic mandatory requirement for the manufacturers to commit to is the Just-in-Time (JIT) concept or time-based manufacturing (Simkins and Maier 2004). This will ensure that the goods are manufactured, produced and assembled by the manufacturing company according to the given specifications of the order received from the customers. The manufacturing company can carry out these steps of production and produce the goods in its own manufacturing plant(s), which we term in-house production; or contract the production to a third-party manufacturing company, which we term out-sourced production. It may also

O. Hussain (✉) · T. Dillon
Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia
e-mail: o.hussain@cbs.curtin.edu.au

T. Dillon
e-mail: Tharam.Dillon@cbs.curtin.edu.au

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_4,

be the case that a manufacturing company utilizes both these types of production to obtain the final end product(s) according to the customers' order and specifications within the specified time frame. But whatever the production mechanism utilized, the manufacturer must ensure that the output generated by its manufacturing plant(s) must be proportional to the real-time demand of the customers. We characterize the 'demand' by the total revenue that would be generated by fulfilling *all* the customer orders within the required time frame. This ensures that all the required product(s) will be available to be delivered to the consumer(s) within the specified period of time.

But due to various factors/uncertainties, the manufacturer might not be able to produce the required end product/s as desired by the customers within the specified period of time. This will have serious implications for the manufacturer both financially and in terms of its reputation. In order to avoid such implications, some important analyses for the manufacturers to consider in real-time production systems are:

1. Whether they have the production, machine capability and capacity by which they can commit to the demand of customers within the specified period of time.
2. Whether other uncertain factors exist in order to determine beforehand the probability of their not achieving the expected demand within a specified period of time.

Analysis of these factors would assist the manufacturer to determine the unserved demand from its production units and to develop alternate strategies by which it commits to the expected demand of a time period and alleviate any losses. In this chapter, we extend our previous work and propose a cost-benefit analysis for choosing third-party producers as an alternate strategy for meeting the unserved demand of a time frame. The chapter is organized as follows. In Sect. 4.2 we give a brief overview of the related work from the literature and then discuss briefly our previous work by which the manufacturer takes into consideration the uncertainty inherent to (a) consumer's future demands (b) possible plant failure and (c) availability of inputs; and determines the probability of it not meeting the required demand of the time period. In Sects. 4.3 and 4.4 we develop an approach by which the manufacturer does a cost-benefit analysis for deciding whether to hedge the production of the unserved demand with third-party manufacturers to meet the total demand of a time frame. Section 4.5 concludes the chapter.

4.2 Related Work

Existing work in the literature can be classified into 3 different categories that aim to improve the production efficiency in demand-driven production. We classify the first category of those approaches as the preproduction process which is the one before the raw materials are fed into the production units in order to obtain the end product (Mohebbi et al. 2007; Zäpfel 1998; Chen et al. 2004; Yıldırım et al. 2005; Qiu and Burch 1997). The second category is termed the postproduction process which is

the phase after the goods come out of the production units (Tan and Gershwin 2004; Zhang et al. 2009; Sharma 2009; Wu 2009). However, prior to this, there is one more important phase which is the actual production process, when the goods are being manufactured by the production units. The successful completion of this phase is very important, as it will ensure that the required goods are manufactured within the required time frame for the post-production process to start. But there might be various uncertainties at this stage that may result in the goods not to be manufactured in the required time frame. Some of them to mention are (a) the inability of the manufacturing units in the manufacturer's various production plants to meet the required demand due to their unexpected outages and (b) the unavailability of raw materials in the required quantity at the required time. The identification of such uncertainties and analysis of their effects are important to be considered for developing appropriate risk mitigation policies and to keep the production schedule on track. Our previous work takes into consideration the stochastic nature of consumer's demands over a given time period and determines the impact of the uncertainty of (a) plant failures and (b) availability of inputs, to ascertain the probability of the manufacturer not meeting the required demand by its production units. We will explain it briefly in this section.

It is highly likely that the customers' demand will fluctuate over a given period of time (Berndt 2006). To capture and model that accurately, we adopt the methodology proposed by Chang et al. (2006) and break up the total time period of consideration, termed as the time space into different subintervals of time. Each subinterval of time is termed as the time slots. The expected demand in each time slot of the time space might vary according to the real-time customers' orders in it and the other operational costs. We capture such variedness of demand in a given time period by plotting the *Demand Expected Curve (DEC)* of the time space. The DEC, as shown in Fig. 4.1, represents the accurate probability of an amount of revenue to be produced by the manufacturer from its production units over the time space. This amount is a combination of the expected customers' demand and the operational costs that it will incur in each time slot of the time space. The DEC is plotted by determining the cumulative probability to achieve a financial amount of revenue in each time slot of the time space, as shown in (4.1):

$$P(x) = \frac{m}{n} \tag{4.1}$$

where: $x =$ the financial amount on which the Demand Expected Curve is being determined, $m =$ the number of time slots in which the financial demand 'x' has to be at least achieved, $n =$ total number of time slots in the time space.

However, due to the uncertainties of its production units' unscheduled downtime and/or nonavailability of the required raw materials, there is the possibility that the manufacturer will not achieve the required demand of the time space as represented in the DEC. This will result in having unserved demand in the time period. To model that, we proposed that a multistate probabilistic discrete function as shown in Fig. 4.2 be utilized to capture the different levels of uncertainty from these factors. This function will capture the various levels of deviation from the optimal points of

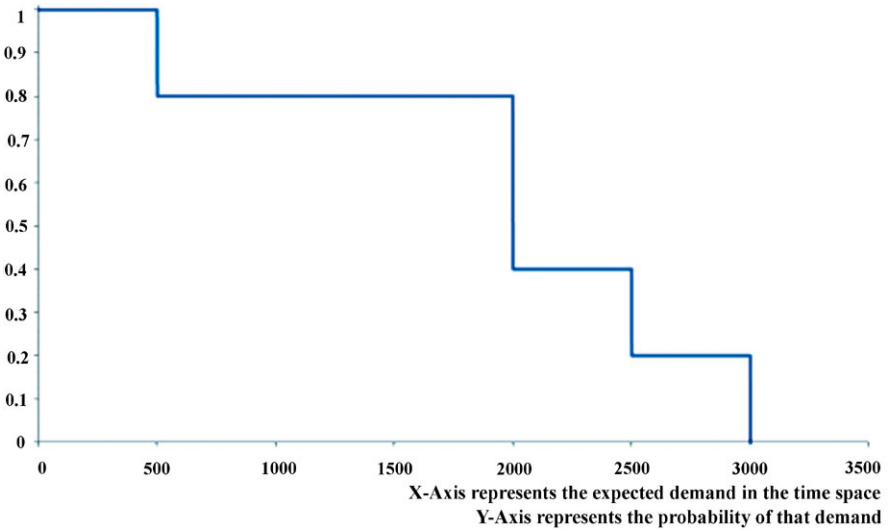


Fig. 4.1 The DEC according to the expected demand during the time space

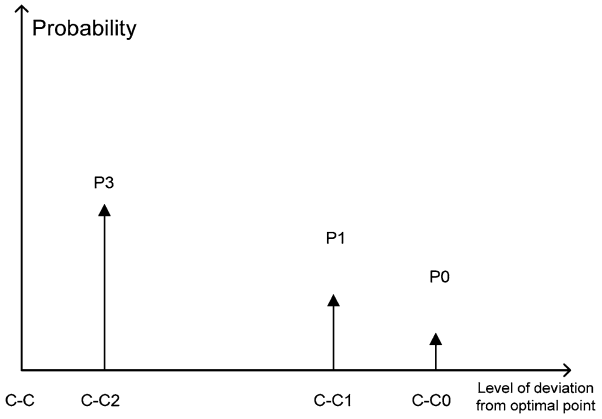


Fig. 4.2 The uncertainty in the availability of raw materials or production units during the time space

each uncertain variable along with their probability of occurrence. To determine the impact of such uncertainties on the expected demand to be met, we proposed the calculation of the *Equivalent Demand Expected Curve (EDEC)* of the time space as shown in Fig. 4.3.

The mathematical operator convolution is utilized to determine the impact of the uncertainties on the DEC as shown in (4.2):

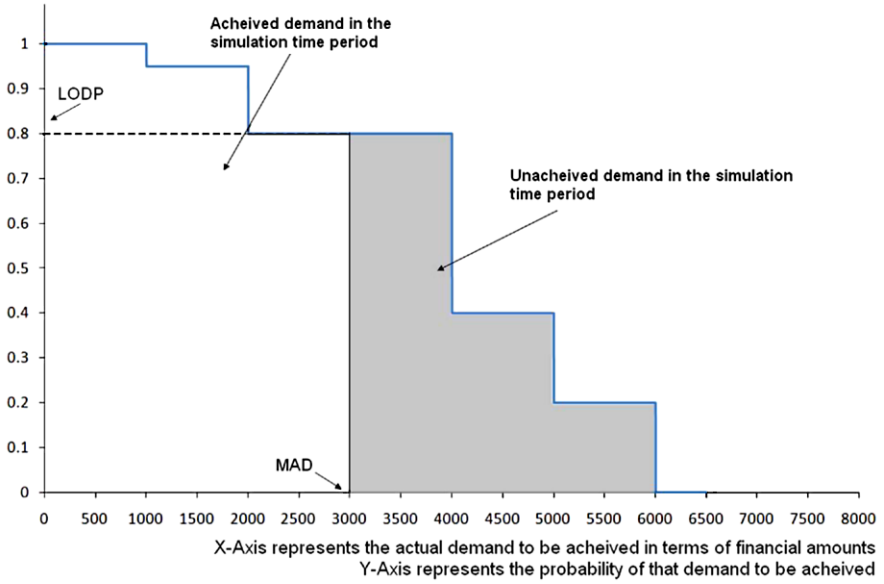


Fig. 4.3 The EDEC for the duration of the time space

$$\text{EDEC}(x) = \sum_{i=1}^{n} p_i * \text{DEC}(x - \text{FOR}_i/\text{URM}_i) \quad \text{for } (x - \text{FOR}_i/\text{URM}_i) \geq 0$$

or

$$\text{EDEC}(x) = \sum_{i=1}^{n} p_i \quad \text{for } (x - \text{FOR}_i/\text{URM}_i) < 0 \tag{4.2}$$

where: $n =$ the number of outage levels of a production unit or the number of levels of unavailability in the raw materials, $x =$ the point or an amount at which the EDEC is to be ascertained, $\text{FOR}_i =$ degree of severity of the outage level i, $\text{URM}_i =$ level i of unavailability of the raw material, $p_i =$ magnitude of occurrence of level i, $\text{DEC}(x - \text{FOR}_i/\text{URM}_i) =$ the level of demand expected at point $(x - \text{FOR}_i)$ and $(x - \text{URM}_i)$.

As compared to the DEC, the EDEC has increased probability of the expected demand of an amount to be met within the time space by considering the different types of uncertainties. This modification reflects the fact that when some production unit(s) are randomly forced offline, the remaining production unit(s) should see a backlog of goods to be produced, to pick up and compensate for the demand that is being unserved by the machines which partially or totally are forced off-line, or when some required goods are falling short, then the backlog is determined and taken into consideration. To determine the financial loss due to such uncertainties, the manufacturer should ascertain the:

- Loss of Demand Probability (LODP);
- Un-served demand by its production units.

LODP represents the probability of the manufacturer not meeting the required demand from its production units in the required time frame. The LODP index of a time period as shown in Fig. 4.3 is the ordinate on the EDEC corresponding to the maximum amount of generated revenue (MAD) by the manufacturer from its production units on the abscissa. The 'Unserved Demand' of the time space represents the additional demand which has to be generated and met by the manufacturer from the maximum achieved demand of its production units. But this level(s) of demand will not be generated and met by the manufacturer within the time space as it is beyond the capacity of its production unit(s) and can be termed as the financial loss that it can experience.

To minimize such losses, the manufacturer has to develop strategies by which it can commit to or achieve these levels of unserved demand in the time space. One such strategy is to anticipate and store the required parts or products according to a set of specifications in its inventory. But this may lead to inventory paradox whereby the manufacturer does not clear the stored stock in a timely manner. An example of this is the Amdahl Computer Corporation which held an inventory of all possible configurations, leading to hundreds, if not millions, of dollars of excess inventory (Lebovitz and Graban 2001). Another strategy which the manufacturers can adopt in such situations is to hedge the production of the required unserved demand with third-party producers, whereby it will achieve the total expected demand of the time space. In this chapter, by 'hedging' we mean the process of utilizing third-party producers for manufacturing the goods and paying them for their services. But before utilizing this strategy, a cost-benefit analysis has to be carried out to determine whether it is feasible or profitable for the manufacturer to hedge with the third-party producers. Such an analysis is important and will have significant impact on its decision-making. In this chapter we will propose such a methodology for doing a cost-benefit analysis for decision-making.

4.3 Cost-Benefit Analysis to Hedge with Third-Party Producers

In the cost-benefit analysis, we aim to analyze and compare the cost that will be incurred by the manufacturer to hedge the goods from third-party producers with that of the loss that would be incurred by it by not meeting the required demand of the time period. An important point to consider here is that such an analysis has to be done only on the unserved demand of the time period, which is the part of the EDEC after the MAD in Fig. 4.3. Taking that part of the EDEC, the manufacturer should first determine in crisp financial amounts the unserved demand. To achieve this, we propose that the manufacturer should determine the probability mass function of the EDEC. Doing so would give the probability to which each level of amount has to be generated by the production unit(s). To consider the unserved demand of the time space, the manufacturer should consider the part of the EDEC after the MAD and then utilize the probability of each level of unserved amount to determine the number of time slots in which that level of amount had to be achieved within the

time space. Based on the determined analysis, the manufacturer can ascertain the total crisp unserved demand within the time space.

For example, let us consider the EDEC as shown in Fig. 4.3 where the MAD from the production unit(s) in the time space is $3,000. In order to decide whether to hedge with third-party producer(s) to meet the total demand, the manufacturer needs to first determine in crisp amounts the level of unserved demand of the time space. Determining and representing in Fig. 4.4 the probability mass function of the EDEC shown in Fig. 4.3.

To determine the crisp amount of unserved financial amount, the manufacturer by utilizing the probability of each amount on the EDEC from the MAD should determine the number of time slots where such level of loss is experienced. In the current example, we consider that there are five time slots in the simulation time period. Figure 4.5 represents the occurrence of each level of loss from the MAD in the time slots of the time space. From Fig. 4.5, the crisp amount of loss that could be incurred by the manufacturer during the time space is $7,000.

By considering the downtime of the manufacturer's production units and the uncertainty of the availability of the raw materials as shown in Fig. 4.2, the manufacturer can determine the quantity and type of products that it will not be able to achieve from its production units within this time period. Utilizing such information, the manufacturer should determine the costs that would be incurred to it in hedging those products with third-party producers. It can then do a cost-benefit analysis by comparing the costs of hedging the production of the goods with the loss that it would otherwise experience by not committing to the demand and then decide whether to consider this option as an alternative. If the cost of hedging the goods from third-party producers is less than the level of loss that would be incurred by it, then the manufacturer can consider this strategy as an alternative to achieve its total demand of the time space. On the other hand, if by analysis the manufacturer determines that the level of loss that it could incur is less that what it would spend to hedge the goods from third-party producers, then it need not consider this as an

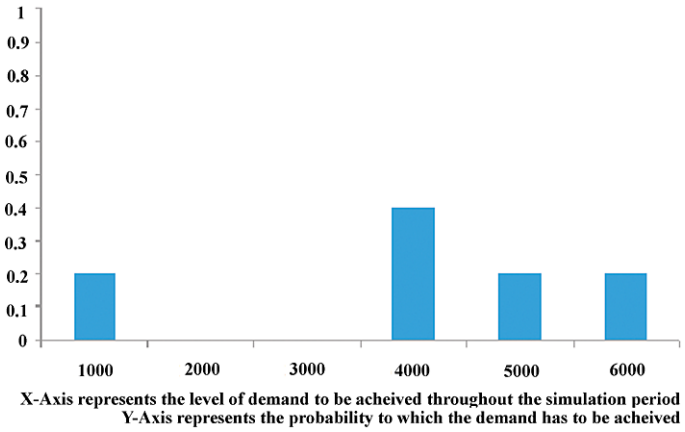


Fig. 4.4 The probability mass function of the EDEC

alternative for achieving its expected demand unless supply to a customer has high business strategic significance, e.g., a very large and important customer with which the manufacturer would not like to spoil its reputation.

Extending the previously discussed example, let us consider that the manufacturer, by analyzing the operational history of its production units and uncertainty in the availability of raw materials, determines that it has to hedge the production of Product 1, Product 2, and Product 3 from third-party producers in order to meet its total demand. Let us consider that the manufacturing and operational costs of the third-party producer depend on the quantity of each product to be manufactured are as shown in Table 4.1. The total cost for the manufacturer to hedge with third-party producers from the quantity and costs mentioned in Table 4.1 is $6,800. The manufacturer, by comparing it with the loss that it would incur, can make an informed cost-benefit decision to consider this approach as one of its alternatives to achieve the total demand of the time space.

In some cases, the proposed approach will not give the manufacturer the accurate number of time slots in which it will experience a loss of a financial amount. Without such an analysis, it may not be able to determine the cumulative loss due to the unserved demand of a time period. In such scenarios, we propose that a probabilistic approach be utilized to carry out the cost-benefit analysis. We explain that approach in the next section.

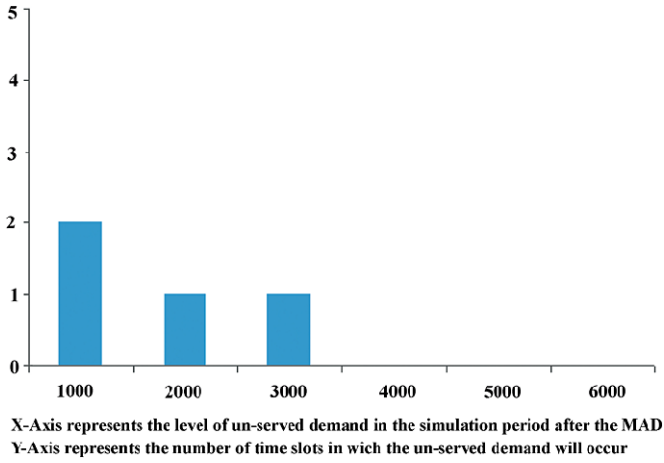


Fig. 4.5 Occurrence of the level of unserved demand in the number of time slots of the time space

Table 4.1 Operational costs of manufacturing the goods from third-party producer

	Product 1	Product 2	Product 3
Quantity required	1000	100	200
Cost per quantity	$ 0.50	$ 20	$ 10
Misc. Costs	$ 0.30 per piece	$ 0	$ 2000
Total	$ 800	$ 2000	$ 4000

4.4 Probabilistic Approach for Cost-Benefit Analysis to Hedge with Third-Party Producers

We propose that the manufacturer by utilizing the part of the EDEC after the MAD should determine the accurate probability of experiencing different levels of financial loss amounts within the time space. Based on the assessment, it should plot a probabilistic curve that represents the different level(s) of expected loss due to the unserved demand of the time space. In order to produce such a curve, the manufacturer from the EDEC should determine the individual probability of occurrence of the different levels of unserved demand and then normalize them so that their cumulative probability is equal to 1. In other words, the cumulative sum of the probability of occurrence of each level of resource on the EDEC after the MAD should satisfy the condition in (4.3):

$$\sum_{n=1}^{z} p(x_n) = 1 \tag{4.3}$$

where z represents the number of levels on the EDEC after the MAD, x_n represents each level of resources after the MAD on the EDEC, and $p(x_n)$ represents the probability of occurrence of the level of resource.

In order to explain with an example, let us consider the EDEC as shown in Fig. 4.3 and the manufacturing scenario discussed in the last section, where the manufacturer wants to perform a cost-benefit analysis in order to determine whether or not to hedge with third-party producers so as to meet the unserved demand. Let us assume that the maximum achieved demand (MAD) is $3,000. The probability mass function of the EDEC after the MAD is represented in Fig. 4.6.

In order to obtain a probabilistic curve that represents the level of unserved demand, the manufacturer should normalize the probability mass function shown in Fig. 4.6 so that it satisfies Eq. (4.3). Based on the normalized values, the manufacturer can plot a probabilistic curve which represents the probability of occurrence of each level of unserved amount(s) in the time space. This curve represents the level of loss that the manufacturer can experience due to the unserved demand. We term such a probabilistic curve as the 'Expected Loss Curve' (ELC). Figure 4.7 represents the ELC from the above example.

To perform a cost-benefit analysis, the manufacturer should compare the level of loss that it could experience due to unserved demand, with the cost of alleviating this by third-party producers. As mentioned earlier, by utilizing the expected downtime of the production units and the uncertainty in the availability of raw materials, we consider that the manufacturer knows the various product types and their quantity for which it needs to depend on other producers. Utilizing this, the manufacturer should determine the cost that it would incur when hedging the production of the goods with the third-party producers, in each time slot of the time space.

Based on this analysis, we propose that it plots a probabilistic curve, 'Cost of Hedging Curve' (CHC) that represents the cost of hedging to meet the unserved demand of the time space. In order to perform a cost-benefit analysis, the manufacturer should compare the area of the Cost of Hedging Curve with the Expected Loss

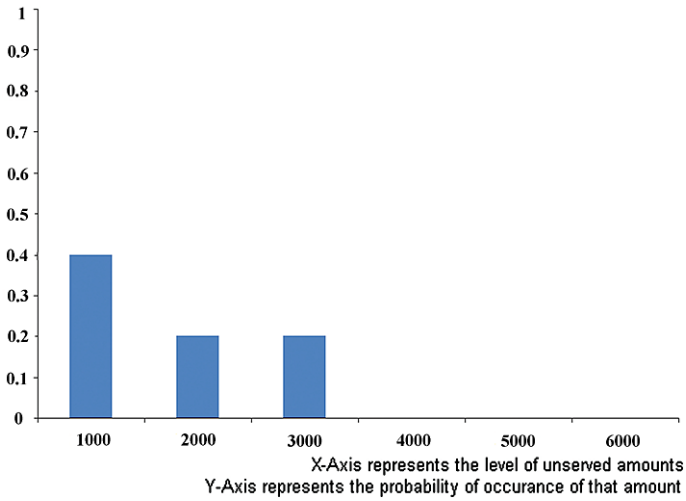


Fig. 4.6 Probability mass function of the EDEC after the MAD

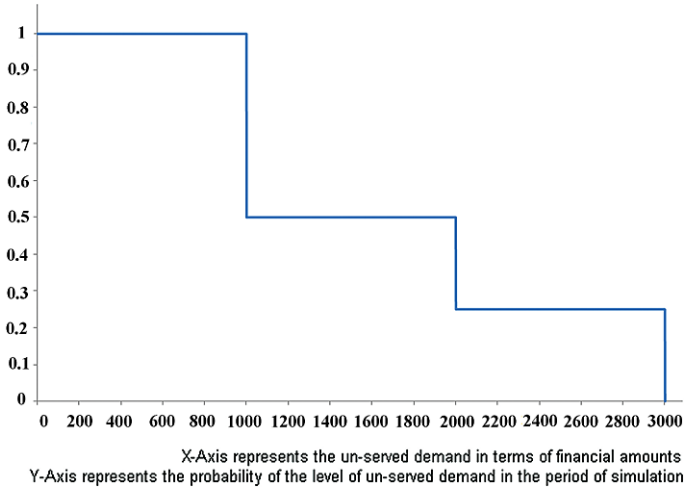


Fig. 4.7 Probabilistic curve showing the level of unserved demand in the time space

Curve which is determined by (4.4) and (4.5), respectively:

$$\text{Area of Expected Loss Curve} = \int_0^n \text{ELC} \tag{4.4}$$

where n represents the amount till the ELC extends.

$$\text{Area of Cost of Hedging Curve} = \int_0^m \text{CHC} \tag{4.5}$$

where m represents the amount till the CHC extends.

If ELC > CHC, then the costs which the manufacturer would incur from third-party producers to meet the unserved demand is less than the loss that it would experience as a result of that unserved demand. In such cases, the manufacturer can consider hedging with third-party manufacturers as one of its alternatives to meet the total demand for the time space. If, on the other hand, CHC > ELC, then it means that the costs that it would incur to alleviate the loss as a result of unserved demand from third-party manufacturers is more than the loss that it would actually experience due to the unserved demand. In such cases, utilizing such an alternative of hedging the goods with third-party manufacturers is not a feasible alternative for the manufacturer in order to achieve the total demand of the time space.

To explain with an example, let us consider the previously discussed scenario of the manufacturer trying to determine whether or not to hedge with third-party producer(s), to achieve the total demand of the time space. The probabilistic curve showing the unserved demand of the time space is shown in Fig. 4.7. Let us consider that the costs of hedging with third party manufacturers in each time slot of the time space is as shown in Table 4.2.

Based on the total amount to be incurred in each time slot, the Cost of Hedging Curve (CHC) is as shown in Fig. 4.8. By utilizing (4.4) and (4.5), the manufacturer should determine the areas of the ELC and CHC, respectively, and compare them. Comparing the ELC and the CHC curves of the present example and representing these in Fig. 4.9.

As seen from Fig. 4.9, the cost of hedging the goods from a third-party manufacturer will be less than the loss that a manufacturer would experience as a result of not achieving the unserved demand, and the manufacturer can hedge with third-party producer(s) in order to meet the required demand of the time space. By utilizing the proposed approach, the manufacturer can make an informed decision about whether or not to hedge with third-party producers, in order to meet the total required demand for the time space. It can then take appropriate decisions to collaborate with different third-party producers so that it can increase its production capability enabling it to commit to the expected demand of a given time frame period.

Table 4.2 Operational costs to hedge with third-party manufacturer in each time slot

For	Manufacturing costs in				
	Time Slot T1	Time Slot T2	Time Slot T3	Time Slot T4	Time Slot T5
Product 1	200	200	100	200	100
Product 2	500	500	500	500	0
Product 3	1000	1500	1000	0	500
Total Costs	1700	2200	1600	700	600

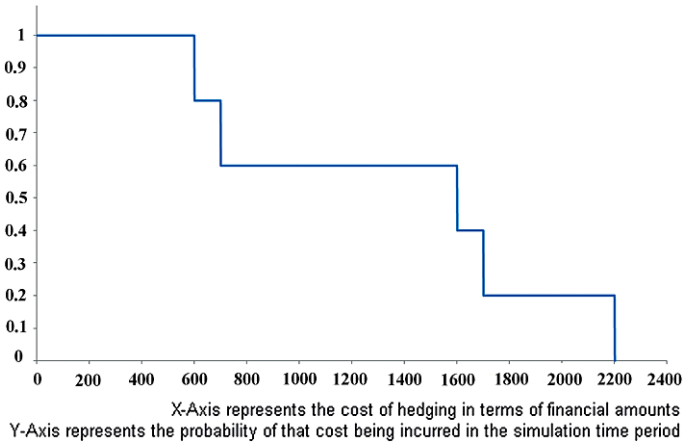


Fig. 4.8 The CHC to meet the unserved demand

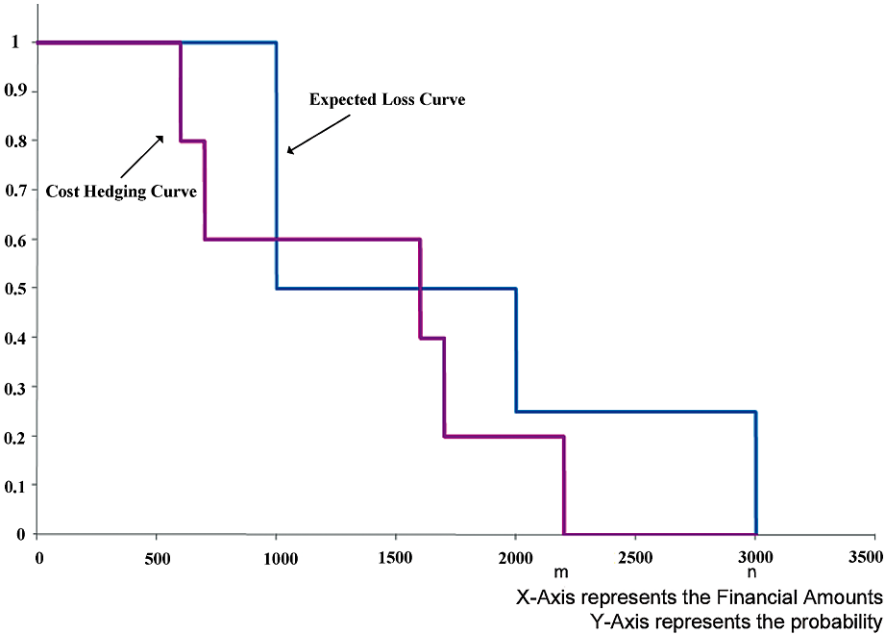


Fig. 4.9 Comparing the CHC and ELC to meet the unserved demand of the time space

4.5 Conclusion

In this chapter, we propose an approach by which a manufacturer in demand-driven production can make an informed cost-benefit analysis decision to decide whether or not to hedge with third-party producers, in order to meet the required demand

of the time space. By required demand we mean the total revenue that would be generated by fulfilling all the customers' orders in the required time frame. The approach that we propose determines, in (a) crisp amounts and (b) probabilistic terms, the cost that the manufacturer would incur in hedging the goods with the third-party producer(s) over a given time frame. It can then compare this analysis with the loss that it would experience due to the unserved demand in that time frame and then make an informed decision. The solution to the problem will be of primary interest to the manufacturer or the revenue planner in manufacturing plant(s) as it should guide him/her to make accurate decisions on operational matters by which it can achieve the required demand of the time space.

References

B. Berndt: Accurate planning in demand-driven world. Quality Control: Manufacturing & Distribution (2006) 56–58

E. Chang, T. Dillon, F. K. Hussain: Trust and Reputation for Service-Oriented Environments. Wiley, West Sussex (2006)

M. Chen, R. Dubrawski, S. P. Meyn: Management of demand-driven production systems. IEEE Transactions on Automatic Control **49** (2004) 686–698

R. Lebovitz, M. Graban: The journey toward demand driven manufacturing. 2nd International Workshop on Engineering Management for Applied Technology (2001) 29–35

E. Mohebbi, F. Choobineh, A. Pattanayak: Capacity-driven vs. demand-driven material procurement systems. International Journal of Production Economics **107** (2007) 451–466

M. M. Qiu, E. E. Burch: Hierarchical production planning and scheduling in a multiproduct, multimachine environment. International Journal of Production Research **35** (1997) 3023–3042

S. Sharma: Revisiting the shelf life constrained multi-product manufacturing problem. European Journal of Operational Research **193** (2009) 129–139

S. Simkins, M. Maier: Using just-in-time teaching techniques in the principles of economics course. Social Science Computer Review **22** (2004) 444–456

B. Tan, S. B. Gershwin: Production and subcontracting strategies for manufacturers with limited capacity and volatile demand. Annals of Operations Research **125** (2004) 205–232

C.-W. Wu: Decision-making in testing process performance with fuzzy data. European Journal of Operational Research **193** (2009) 499–509

I. Yıldırım, B. Tan, F. Karaesmen: A multiperiod stochastic production planning and sourcing problem with service level constraints. OR Spectrum **27** (2005) 471–489

G. Zäpfel: Customer-order-driven production: an economical concept for responding to demand uncertainty? International Journal of Production Economics **56–57** (1998) 699–709

Q. Zhang, M. A. Vonderembse, M. Cao: Product concept and prototype flexibility in manufacturing: Implications for customer satisfaction. European Journal of Operational Research **194** (2009) 143–154

Chapter 5
A Security Assurance Model to Holistically Assess the Information Security Posture

Igli Tashi and Solange Ghernaouti-Hélie

Summary Managing *Information Security (InfoSec)* within an organization is becoming a very complex task. Currently, InfoSec Assessment is performed by using frameworks, methodologies, or standards which consider separately the elements related to security. Unfortunately, this is not necessarily effective because it does not take into consideration the necessity of having a global and systemic, multidimensional approach to ICT Security evaluation. This is mainly because the overall security level is only as strong as the weakest link. This chapter proposes a model aiming to holistically assess all dimensions of security in order to minimize the likelihood that a given threat takes advantage of the weakest link. Then a formalized structure taking into account all security elements is presented. The proposed model is based on, and integrates, a number of security best practices and standards that permit the definition of a reliable InfoSec framework. At this point an assessment process should be undertaken, the result of which will be the assurance that InfoSec is adequately managed within the organization. The added value of this model is that it is simple to implement and responds to concrete needs in terms of reliance upon efficient and dynamic evaluation tools and through a coherent evaluation system.

5.1 Why Is Information Security Assessment a Complex Task?

One of the major issues in security evaluation is finding a balance between two commonly used extremities, such as checklists and auditing. Obviously, these two means of evaluating InfoSec have their own advantages, which is not the subject of this chapter. A lot of work has been done in this field and many methodologies and frameworks developed to perform security assessments based on both, checklists

I. Tashi (✉) · S. Ghernaouti-Hélie
Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland
e-mail: igli.tashi@unil.ch

S. Ghernaouti-Hélie
e-mail: sgh@unil.ch

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_5,

and audits. The objective of the model proposed in this chapter is to *fill the gap between these two extremities and to propose a way of assessing InfoSec which goes into more depth than a simple checklist but at the same time does not demand as many important resources as the audit*.

5.1.1 Traditional Assessment Procedures

Notwithstanding their advantages, security checklists and InfoSec audits remain a static way of evaluating the security. Moreover, security checklists are often directly extracted from audit frameworks or methodologies as a simplified tool for demonstrating the degree of conformity with the framework's requirements. The output of the checklists will be a state of security that corresponds to the requirements of the standard (or the framework), but not necessarily to the needs of the organization. By their very nature security checklists often do not embrace detailed analysis, as commonly responses such as "yes" or "no" are sufficient. In this context way, the relationships between different control processes or security measures are difficult to detect, and a number of significant elements may be missed from the conclusions. The same criticism can be applied to audit activities which are performed with a higher degree of granularity. Security audits commonly do not have the explicit objective of assessing and evaluating the security level of the organization, but rather of verifying that the internal control system provides indirect assurance over the reliability of the organization's financial statements. The mission of InfoSec goes beyond this, by providing assurance that a range of risks and their impact have been considered. This is not only in terms of information reliability but also in terms of its availability and confidentiality and other security criteria. In our opinion, a deep understanding of the organization which is the subject of the audit, instead of a deep understanding of the methodology itself, is required when performing an assessment of InfoSec. Based on the degree of granularity and the audit effort necessary, it is nearly impossible, or at least task requiring great resources, to perform audits capable of covering all security concerns during a single period. Consequently, and under the pressure of being as exhaustive as possible, audits concerning different security domains are performed in different periods. Another potential gap is that audits are performed more or less periodically, often every six months or a year, and thereby do not address the continuous nature of security. By this we do not mean that audit has no impact on the security level of an organization, but rather that within *audit activities, compliance and conformity goals prevail over security optimization*. Therefore the holistic view, which might be a good basis to inspire trust in the security system in place, is not achieved. Evaluation activities such as checklist and/or audits respond more to the "what" related issues rather than the "how" related issues.

Overall, we can also comment that the certification of compliance with standards often follows the same path as general audits, in that they may concern very specific and technical aspects of security. We miss here the holistic requirement to consider the security system as a whole. Even dedicated standards, as, for example, ISO

27001:2005 (ISO-Std. ISO/IEC 27001:2005 (E) 2005), do not permit the evaluation InfoSec as a holistic process, as they are limited to the architectural characteristics. At the same time, considering the fact that a procedure of standardization is hidden behind each standard, it should be remembered that, nevertheless, the primary goal remains the degree of conformity with respect to the standard. It has to be underlined that the organizations following and tracing the guidelines for standards will still have to provide efforts in terms of adjustments regarding their specific activities.

5.1.2 New Challenges Regarding the Assessment of InfoSec

The ambition of being effective and efficient demands a deep knowledge of the organization and its business objectives in order to establish not only *a coherent InfoSec System* but also *a coherent Assessment System*. Based on this deep knowledge of the organization, security objectives have to be based on the security needs of the organization. The InfoSec System in place has to fulfill a group of requirements resulting from some objectives for protection that are attainable and sensible. This becomes even more important when considering the fact that the adherence of employees is a condition of success for InfoSec.

Let us take the example of a widely used Best Practice Standard concerning InfoSec, namely ISO 27002:2005 (ISO-Std. ISO/IEC 27002:2005 2005), and ask the questions: Does ISO 17799:2005 allow organizations to reach a certain security level, and is this the more appropriate level for the organization? There is no doubt that the Standard is always of great utility. But how do the ISO's standards requirements match those of a specific organization? And what about the financial capacities an organization should possess in order to maintain the requirements of the standards?

Informational risk and the way to assess it is a matter of perception and having a good level of InfoSec management will depend on this perception. In general, satisfying the requirements of ISO 17799:2005 might show due diligence but do not show the effectiveness of any outputs. This standard is a checklist of procedures and controls to be implemented, with an emphasis on conformity. This affirmation reveals an *important gap* introduced by the use of, and the importance imputed to, standards: *the misunderstanding of the fact that standards are dedicated to a customized scope and remain a baseline approach*. Although standards and best practices provide an important component of designing security, organizations should not rely upon them blindly (Mercuri 2003).

Another InfoSec related standard, ISO 27001:2005 (ISO-Std. ISO/IEC 27001: 2005 (E) 2005), provides a model for establishing, implementing, and maintaining an InfoSec Management System. The question to be asked is: Does this ISO 27001 conformity ensure that the existing security management system works in the best possible way? Conforming to a standard is indeed a good starting point. Many commercial consulting companies selling ISO conformity exist in the market, demonstrating that this is apparently a good business. But there is an ambiguity about the "conformity" they sell. In fact, they often sell advice on developing security policies

or evaluating the security processes of the organization and their compliance with the ISO security standards framework. It is essentially a form of consulting work which treats security issues in a very generic and uniform way.

Security managers need to produce effective security, as following standards by themselves will not be sufficient. ISO standards give directives but do not specify either their effectiveness or how a given security level can be achieved. A larger set of tools is needed in order to achieve the principal goal that is a higher and appropriate security level. The driver of InfoSec is to build confidence into ICT infrastructures. Being in conformity with a standard certainly brings some added value, but does it provide confidence over security (Tashi and Ghernaouti-Hélie 2006; Tashi and Ghernaouti-Hélie 2007)?

This is another question to which we have to respond when we consider an InfoSec Management process. Expressing confidence by conformity is not sufficient; it must be linked to the quality of the system. Satisfying the requirements of ISO 27002:2005 and ISO 27001:2005, in our opinion, does not show the way to building a proper the management system. Complying with a standard does not mean going into depth. As we have already mentioned, InfoSec is an ongoing process realized in a dynamic environment, while conformity (or certification when it does exist) is only valid for a static state of a process or component. Conformity does not integrate the mandatory and anticipatory dimension of InfoSec required by the evolving nature of IT risks (Tashi and Ghernaouti-Hélie 2009). At present, the trend is to state that the organization is appropriately managed if it complies with specific regulations and standards. It is mostly a legal protection, but it does not produce an effective InfoSec. Legal and regulation constraints should be seen as a key factor to oblige the organization to put in place an effective InfoSec management framework. For doing that ISO Standards could help, but one should keep in mind that an organization's competitiveness depends on its InfoSec effectiveness and conformity is not the only element which guarantees security effectiveness.

5.1.3 A New Multidimensional InfoSec Assessment Framework

As has been indirectly suggested several times during this introduction, the aim of the assessment model is to *fill the gap left by several evaluation methods currently and commonly utilized, whilst being less time consuming and labor intensive.* Our driver was the utility of such methods from a user point of view, namely internal stakeholders, business partners, and other external parties. *The objective is to produce a conceptual framework in order to holistically evaluate the InfoSec System of a given organization, producing a certain level of confidence and trust in the protection practices operated within the InfoSec System.*

InfoSec is a discipline including several dimensions and conforms to the principle that *the security of a given system is only as strong as its weakest link*. The weakest link might be located within any of the security domains, so that, each of the domains concerning InfoSec must be evaluated and assessed using formalized methods.

In order to be put into practice, the model is based *on an initial assumption* that security attributes such as the following exist beforehand:

- Basic InfoSec issues have been considered and are defined within the organization;
- An Information Security Management System (ISMS), or something similar, already exists;
- The organization has already a security structure in place and already possesses established security policies and procedures;
- Security dimensions such as those that are described within the model, namely organizational, operational, human, and legal, are in place;
- Security resources (hardware and software) have been installed, configured, and maintained (Butler 2002);
- Some measurement of efforts and outputs is provided in terms of InfoSec effectiveness and efficiency.

The conceptual model presented within this chapter will consider InfoSec from three different points of view, see Fig. 5.1:

1. The first one is InfoSec as a Program (or System). InfoSec must include several activities jointly performed in order to obtain a security posture corresponding to the business requirements. Indeed, the InfoSec posture is the state of protection that results from general and particular measures taken, formally or informally, to ensure security objectives (Barafort et al. 2006). The structure responds to three main characteristics: *completeness, capability,* and *coherence*;
2. The second one is InfoSec as a Process. After having evaluated the architecture and the structure of the InfoSec system, it can be confirmed that some of the baseline conditions to be successful do already exist within the organization. The challenge or objective of this second phase will be to implement and operate

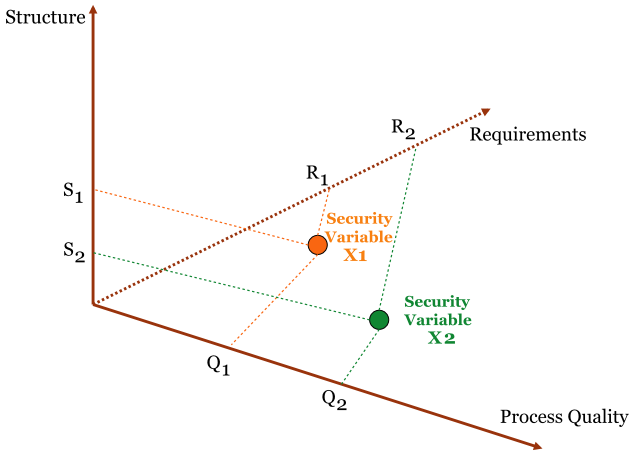


Fig. 5.1 Multidimensional view of InfoSec level

the inputs of the first phase in the best possible way. The structure responds to three main characteristics: *effectiveness*, *manageability*, and *testability*;
3. The third one is InfoSec as a Business Activity. One of the issues historically related to InfoSec activities is the fact that they have been considered by executive levels as a technical matter and consequently seen as a cost center. Nowadays it is widely accepted that the security activities can provide a competitive advantage. The security provided by the system in place must necessarily correspond to the security needs and security requirements of the organization. The structure responds to three main characteristics: *measurement*, *effectiveness*, and *optimization*.

5.2 Assurance Analysis for an Effective and Efficient InfoSec Assessment Framework

The proposed holistic InfoSec Assessment model has as its objective to *produce a certain level of confidence and trust* leading thus to the concept of Assurance which is the topic of this section.

Security assurance is closely related to the concept of confidence that depends on security related properties and functionalities, as well as on operational and administrative procedures (Beznosov and Kruchten 2004). In the same way, the relationship between the assurance concept and confidence is claimed by (Jelen and Williams 1998), specifying that assurance is a measure of confidence in the accuracy of a risk or security measurement. As can be observed, they specifically base the argument on the assumption that a high assurance level equals a high security level or/and a low risk level. Considering the fact that assurance is closely related to confidence and this one is a difficult rationale to measure, we can argue that measuring the security level means having reached a high assurance level.

Based on the aforementioned arguments, the *cornerstone of the assurance concept* from a security point of view is the *security requirements*. In order to show the evidence, InfoSec requirements should include an ensemble of claims related to each security activity or property according to a *nested structure*, see Fig. 5.2. These claims are supported by evidence taking the form of documentation (ISSEA 2003), or, in a more microscopic view, each claim will involve different subjects and predicates. A subject is categorized into: people, process, environment, technology, and enterprise, while predicates are characteristics of each subject (Williams and Jelen 1999).

In other words, demonstrating assurance, which means that safeguards function as intended, should follow the path below.

5.2.1 Security Assurance Principles

Until now, security assurance has been an engineering concept; as such, it has had to respond to a formal structure as described below and in the same time has had

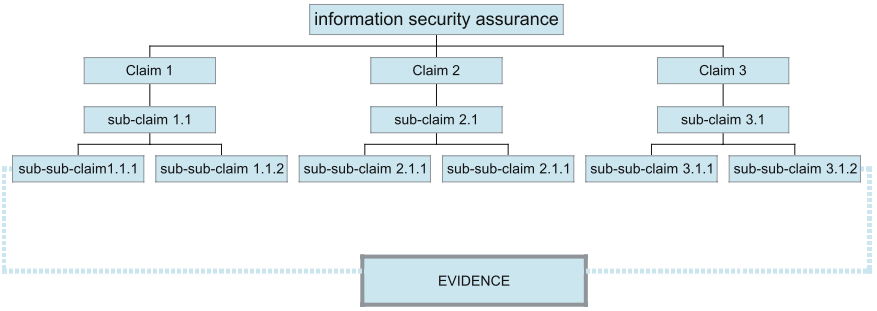


Fig. 5.2 InfoSec assurance nested structure

to be in accordance with some engineering principles as acknowledged in (ISO-Std. ISO/IEC 15408:2005 2006a; ISO-Std. ISO/IEC 15408:2005 2006b; ISO-Std. ISO/IEC 15408:2005 2006c; Merkow and Breithaupt 2005).

In order to be believable and realistic, it has to be accepted that zero risk does not exist and consequently absolute security does not exist either. Implementing safeguards means increasing resistance, thus "buying some time" in order to stop any attack. InfoSec depends on two types of requirements, functional and assurance. This means doing the right things in the right way (Merkow and Breithaupt 2005). In the same way, another important principle when speaking about security assurance is a series of overlapping layers of control and countermeasures providing the assurance that at some point the attack will be thwarted. It is known as the *protection in depth principle*.

Williams et al. (1994) describe a multidimensional structure concerning the concept of assurance itself. They categorize InfoSec assurance into three dimensions, that is to say:

- The *direct dimension* representing a level of confidence on the target of assurance itself;
- The *indirect dimensions* representing the way to examine evidence about the direct dimension;
- The *discrete dimensions* representing elements helping to organize evidence into logical groups.

5.2.2 An Assurance Related Concept: The Trust

It is very important to analyze the trust concept in an InfoSec sphere, since trust is considered as being crucial wherever risk, uncertainty, or interdependence exist. According to the model described in McKnight and Chervany (2000), there are three concepts closely related with trust, namely *the attitude, the belief, and the behavior*. In order to evaluate or assess the security level, these are the three elements capable of holistically addressing InfoSec issues from a high-level view. Based on

this model, there may exist three trust constructions affecting the InfoSec sphere, specifically:

1. The *trusting stance* stating that a better outcome is achieved by dealing with people as though they are well meaning and reliable;
2. The *structural assurance* stating that success is likely because guarantees, contracts, regulations, promises, legal recourse, processes, or procedures are in place;
3. The *trusting belief predictability* stating that if somebody believes that the other's action are consistent enough, then he can forecast them in a given situation.

The question one has to answer when evaluating a subject like InfoSec System is how to demonstrate (in order to gain confidence) that all these requirements about existence of process and procedures, their reliability and predictability are achieved.

An element of response is given by Koskosas (2008), who stated that trust is the ability to effectively understand and communicate the message of security goals. According to Koskosas's definition of trust, a clear *relationship between trust and security requirements* exists. This means that, in order to build confidence in the InfoSec system in place, one should possess some clear specifications of security requirements and adopt a clearly planned and controllable attitude. Evidence should not be provided without some clearly defined objectives. The TCSEC standard (Department of Defense (USA) 1985) formally requests the existence of statements of requirements in order to consider a computer system as being secure. The underlying idea is that there are some components making up a system, likely to meet an advertised purpose.

There are three views to be taken into account in order to understand a complex system and assess its trustworthy character (Slay 2003):

- The *structural view* about the interrelated elements of the system and the way they fit together;
- The *piecewise view* about the identification of the smallest relevant part of a problem;
- The *synoptic view* considering the system as a whole.

While speaking about InfoSec, a distinction has to be underlined between trusted and trustworthy. According to Verbauwhede and Schaumont (2007), a system could be considered as *trusted* when *it provides predictable and reliable behavior*, whereas the trustworthy concept considers the cases where the security policy is not enforced. A system is judged as being *trustworthy* when it *meets the security requirements* that themselves are set up within security policies. Both concepts of trusted and trustworthy are based on the component of *roots-of-trust*.

As a result of these, and in order to gain confidence in the InfoSec system in place, the first action to be taken is to *identify the roots-of-trust of every security related dimension* and then *evaluate every element following the formal structure of the defense in depth schema.*

The paradigm of trusted computed systems holds that trust is a property of a system (Yan and Holtmanns 2007) which could be assessed according to a given

set of standards in some domain of action. If this is true, trust could be formally modeled, specified, and verified. This could be achieved by a formal method of evaluation of InfoSec assurance, as will be demonstrated later on.

5.3 An Holistic InfoSec Assurance Assessment Model (ISAAM)

As we have specified above, the objective of the assessment model was to incorporate a three-dimensional vision of the InfoSec Assessment. Each one of the assessment axes is further described by looking over the most recent publications concerning each one of the evaluation axes and identifying the most important elements explored within the literature.

After that the assessment model as such is presented, based on the analysis that the authors have performed during their research activities.

5.3.1 The Structure

5.3.1.1 Lessons Learned from the Current Methodologies Related to the InfoSec Assurance Structure

The proposed IEEE Information System Security Assurance Architecture (ISSAA) standard specifies the architecture of a systemic approach for managing the state of health of the security controls within an information system without describing the functional characteristics of each component. This security assurance standard fulfills the fundamental requirement to consider the ongoing property of a system, such as InfoSec. The standard complies with the PDCA model recommended by the widespread standard concerning the InfoSec Management System.

The ISSA standard recommends the existence of a Master Control Catalogue composed (MCC) of some *families regrouping some Specific Controls*.

The first component step in the ISSAA standard, see Fig. 5.3, is the Security Categorization. This step concerns a security-focused categorization in order to prioritize the most valuable assets and their protection. This component conveys the same idea of asset inventory required by many other regulations.

After that, the second component, Security Control Selection, is performed implying an *asset categorization* characterized by security objectives (Availability, Integrity, and Confidentiality). This leads to a *security controls categorization* regrouping all the assets responding to the same security objective in order to provide *baseline security controls* corresponding to the first set of assurance requirements.

The other step, named Security Control Supplement Component, aims to perform a risk assessment approach. During this stage, a target set of security controls to be implemented is identified, and variables such as threats, vulnerabilities, and residual risk are assessed.

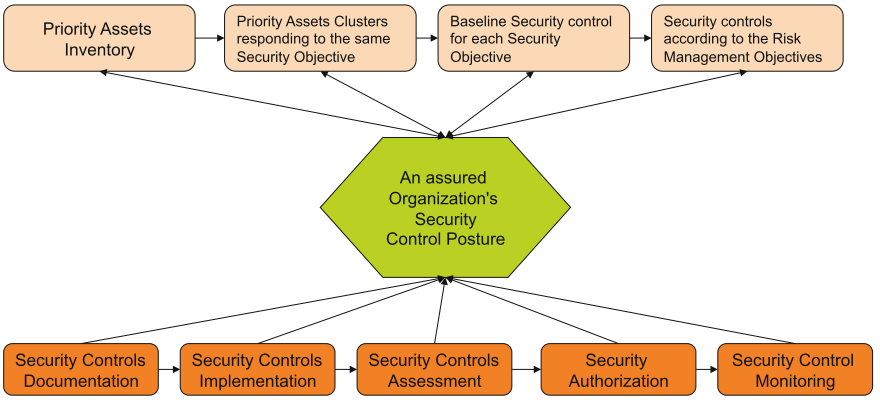


Fig. 5.3 InfoSec functional components (adapted from ISSAA)

Up to this point a security *controls structure and posture* has been defined. This leads to the other components outside the structure focus as Security Control Documentation or the Security Control Implementation. These steps allow gaining results and evidence from tests assuring thus that assurance requirements are met.

The final steps are related to the *assessment of the risk level* after the implementation of the security measures, in order to take the decision while operating or not to end up with a *continuous monitoring* in an improvement scope.

Other standards related to the InfoSec Assurance structure exist within the literature. The evaluation is made considering security mechanisms from a technical point of view in terms of strength, accuracy, etc. The evaluation is of some formal methods based on evaluation criteria in order to show that a security mechanism fulfills all the requirements according to assurance level defined by the standard. That was the case of the European Community and USA through their standards (Department of Defense (USA) 1985; Office for Official Publications of the European Communities 1991). The need to harmonize the evaluation criteria and the procedures to obtain the assurance level has ended in to a widespread certification procedure, called Common Criteria, that has led to the development of a series of ISO international standards, see (ISO-Std. ISO/IEC 15408:2005 2006a; ISO-Std. ISO/IEC 15408:2005 2006b; ISO-Std. ISO/IEC 15408:2005 2006c).

The underlying idea within these different standards is that in order to provide assurance over a security mechanism, procedure, or architecture, generally called a target of evaluation, it has to be *divided into many other components representing given functionalities and expected behaviors*. Functionalities and the expected behavior should satisfy the different security requirements. Assurance is reached when the evidence is given that the security mechanism, procedure, or architecture met the concerned expectancy. Security expectancy is a state of goals moving from a generic or a high level, generally called Security Objectives, to a more concrete level, which are the security requirements.

According to these standards, a security tool, in order to assure the delivery of a secure and reliable service, has to enforce all security requirements constituting the

security target. ITSEC (Office for Official Publications of the European Communities 1991) identifies three security attributes:

- The *security objective*, which represents the functionality the security mechanism is supposed to provide;
- The *security enforcement function,* which represents the functionality the security mechanism must provide;
- The *security mechanism* representing how the functionality is provided.

In addition, ITSEC and TCSEC (Department of Defense (USA) 1985) consider security requirements based on the policy, functionality, effectiveness, and strength. Focalizing the evaluation process on the requirement component underlines the idea that a previous analysis has been performed concerning the security feature, procedure, or architecture under evaluation.

To sum up, assurance arguments should include the following components (Williams and Jelen 1998):

1. Claims, which are the particular property defining the subject of the evaluation and the functional requirements;
2. Evidence, which represents some empirical data contributing to the believability;
3. Reasoning, which will tie the evidence with the claims;
4. Assumption zone, delimiting the space where claims are accepted without evidence

Then the system under evaluation is *divided* into some logical related components and subcomponents in order to provide a formal structure. The formal structure is required to assure that all the components are considered and are subject to the evaluation. Thus, the security subject under evaluation is *divided* into Classes, Families, Components, and elements, as described within the Common Criteria (CC), see (ISO-Std. ISO/IEC 15408:2005 2006b; ISO-Std. ISO/IEC 15408:2005 2006c; ISO-Std. ISO/IEC 15408:2005 2006a). It has to be mentioned that the CC provide assurance through active investigation.

The idea is that IT Security must be integrated from the beginning of the design of a security mechanism. For doing that, the CC proposes two kinds of security requirements, namely *Security Functional requirements* and *Security Assurance requirements*.

The Assurance objective could be achieved by using the following techniques according to CC:

- The mere existence and documentation of necessary and mandatory security processes and procedures;
- Conformity between the processes and procedures in place and the established security objectives;
- Monitoring and tests in order to evaluate the security mechanism, and the analysis of their results.

The Common Criteria define *classes* as a group of families sharing the same focus, *families* as a group of components that share a similar goal but may differ in em-

phasis or rigor, and *components* as the smallest selectable set of elements on which requirements may be based; the element is an indivisible statement of security need.

5.3.1.2 The Holistic InfoSec Assurance Assessment Model (ISAAM) Structure

The InfoSec evaluation will be based on three points of view, see Fig. 5.1 and Fig. 5.4, aiming to provide a final product that will be *the level of trust on the InfoSec System*. This level of trust will *be a general index, which will be based on the perception the evaluator will establish by analyzing the evidence he will gather* during his analysis. By evaluator we mean all interested parties, external or internal, that could be concerned by the security level of the organization.

The model foresees to divide the InfoSec Management System into four principal domains responding to four InfoSec concerns as follows:

1. Organizational Dimension;
2. Operational (Exploited Security) Dimension;
3. Human Dimension;
4. Legal Dimension.

Conceptually, the assessment model identifies the Security Domain as the principal variable to be assessed. Then it is divided into three categories of constitutive components, as shown below.

Model variable → A Security Domain
- Dimension
  - Focus area
    - Specific factors

From a high-level position, each domain corresponds to the same objective that is the risk impact mitigation but uses different procedures or tools to satisfy it. The

Fig. 5.4 A conceptual representation of the approach

underlying idea is that the same risk could take advantage of every existing weakness within the domain and use different breaches to cause harm. On the security side, each domain plays a more or less important role regarding the security objective criteria.[1] As we can notice here, the Risk–Security association is an *"n to n"* relationship, which cannot allow the definition of an exhaustive correlation. For that reason, in order to reduce the complexity, we introduce this metastructure that bounds the capacity to respond to a given risky situation. The organization could prioritize or not one of the domains according to the capacities or resources that currently are or could be available. In that way, instead of evaluating risk exposure, the intention is focused to the preparedness level to face risks.

Each one of the Security Dimensions is then divided into different sub-categories identified as a source of concern within the dimension. A security measure or control will correspond to every source of concern.

Dependant variable → Security Structure (Posture)
Output → Trusted structure of security elements: the key success components regarding security dimensions

Formalizing the InfoSec system's structure allows the basis of the analysis on a solid target of evaluation with some evident elements to be taken into consideration.

A second level of evaluation concerning process quality concerns will then be carried out. The quality concerns of a security process were discussed in the precious section dedicated to the security quality issues. From this analysis five attributes were identified in order to balance each one of the dimension's performance.

Dependant variable → Security Process
Output → Trusted performance of security processes: the key success attributes regarding security dimensions

At this stage we have a well-defined security construct, made up of various components making it possible to mitigate risk impact. Of course, if we base our analysis on a "*cause and effect*" link, we can state that the more security components we put into the system, the better the security level will be, and consequently the higher the trust level. But in the real world, organizations have to work under several constraints, one of them being limited resources. For that reason, they have to identify their security needs and reach a certain security level based on them. To do that, they

[1] By security objective criteria it is intended Availability, Integrity, and Confidentiality.

need to develop a security system which responds to these needs, or try to apply an existing one.

Dependent variable → Security Effectiveness

Output → Trusted accomplishment of security requirements: the key required elements to reach the required security level

Generally, the trust level of an InfoSec System in place will be evaluated based on these interconnected variables as described in Fig. 5.3: in other words, a given architectural structure introduced within a system bordered by the specific security needs.

5.3.2 The Security Quality

5.3.2.1 State of the Art Related to the Quality Issues

ISO 9000:2005 defines quality as the degree to which a set of inherent characteristics fulfills requirements (ISO-Std. 9000:2005 2005). In order to be successfully operated, a process needs to be performed in a systematic manner. The standard defines a process as a *set of interrelated or interacting activities that transforms inputs into outputs*. A quality approach requires taking into consideration the stakeholders' requirements and the definition of the processes, in order to fulfill the requirements and to keep these processes under control. Translating this perspective into an InfoSec Quality (ISQ) oriented assessment, it means that there are three existing dimensions that must be assessed:

- Security Requirements;
- Security Processes;
- Security Controls.

Analyzing the quality concept from a "process approach" point of view, ISO 9001:2000 (ISO-Std. 9001:2000 2000) and (Herrero et al. 2002), emphasizes the *continual improvement* of a quality management system via a PDCA-like model.

In order to provide a quality management system according to ISO 9001:2000, an organization should identify the processes constituting the system and identify the interactions between them. Then it is mandatory to set up criteria for evaluation and ensure monitoring, measuring, analyzing, and continual improving processes. This is because the general concept of ISO 9001 and ISO 9004 is to give the assurance that the customers' needs are met.

According to Ragozzino (1990), quality is the fact *of meeting the objectives and standards in place*. The same idea is found in Feldman (2005) stating that quality is a *measurable meeting expectations and conforming to requirements*. The evidence

of quality is proved by the fact that the design is appropriate, the implementation is careful, and the subject under evaluation meets all the requirements. The quality evaluation of a subject (process, system, procedure) involves performance measurements such as latency, reliability, and availability.

This is why it is very important to add the quality dimension when evaluating the InfoSec posture. It is an upper level of the evaluation, distinguishing organizations with an intrinsic need of integration of security behavior, from those who perform InfoSec as a patch only to ensure compliance with mandatory regulations or laws. The first case should be considered as an organization that has a security attitude, and the second as an organization that does not.

A quality management system is a set of rules or beliefs permitting continually improving performance (Ludwig-Becker 1999). The quality performance includes quality planning, contract review, design, purchasing, training, and servicing.

There are six quality factors identified in Maynard and Ruighaver (2006) addressing security policy issues, that is to say:

1. Functionality—existence of the functions required to perform;
2. Maintainability—ease to maintain the system;
3. Portability—ability to be transferred from an environment to another;
4. Efficiency—amount of resources to perform the required functions;
5. Reliability—capability to maintain performance;
6. Usability—time and resources required to effectively use the subject.

The authors in Gallegos et al. (2004), quoting CobiT, state that quality management is a process that impacts the effectiveness, efficiency, integrity, and availability of Information Systems.

If we opt for security as a process, the attributes that DeLone and McLean propose within their model (DeLone and McLean 1992, 2003) consider success (for an Information System from both points of view, as a system and as a service) as a dependent variable of the:

$$\textbf{Success} = F(\textbf{Accuracy,Flexibility,Reliability,Sophistication,Efficiency,Ease-of-use})$$

In the updated DeLone and McLean Model, another dependent variable is introduced, namely the Service Quality, which holds the following attributes: *Assurance*, *Empathy*, and *Responsiveness*.

Quality Assurance, according to Herrero et al. (2002), involves the commitment of the executive level managers, the establishment of a managerial culture, which means that stakeholders could report the conditions of the risk. In the same way, the main objective of a quality service or process is how to manage nonconformities regarding a given objective and how to control the safeguards supposed to prevent the nonconforming event from occurring. In other words, quality is about documentation, inspection and testing, incident management, crisis management, etc.

Quality assurance comprises all planned and systematic actions necessary to provide adequate confidence that a structure, system, or component will perform satis-

factorily in service (U.S.NRC 2000), based also on the quality attributes proposed by Lee et al. (2002).

Bringing up quality issues related to the InfoSec, the intention is focused into the implementation attribute, and there is a ten-level model based on the ISO 27001 PDCA model proposed in Barafort et al. (2006). Within the Process Implementation Model, the quality of an InfoSec is mainly expressed from an organizational and managerial point of view. When evaluating the quality of an audit process in Karapetrovic and Willborn (2000), the authors enumerate four dimensions of quality. The idea is very interesting, based on the case where the InfoSec will be evaluated from a process-like point of view. The four quality dimensions to take into account when evaluating a process are:

- The management related to the place of the identification of roles and responsibilities within the process under evaluation;
- The planning related to the forward planning conform with a standard or a set of rules;
- The execution related to the conformity of the process to a standard or a set of rules;
- The improvement related to the continuous review and control.

Another well-known standard was studied to understand the security quality related issues, namely the ISF standard (ISF-std. 2007). Quality attributes, such as top management commitment, allocation of appropriate resources, system design to meet requirements, are also analyzed, particularly within the Security Management (SF) and Systems Development (SD) aspects. The ISF standard defines quality assurance as the process that provides assurance that the security requirements are defined, agreed, developed, and met by the system under evaluation.

Another requirement claimed by ISF in the scope of quality was the coordination of security activities in relation to the assignment of responsibilities. Indeed, every system considered as being robust should appoint individual responsibilities. This is in order to ensure that security activities are carried out in a timely and accurate manner (ISF-std. 2007). After that a thorough and regular security audit should be performed. This activity will assess the status of the activity and ensure that security controls are designed effectively. By effectiveness the authors mean the ability of the system to provide the required output (Chamfrault and Durand 2006).

5.3.2.2 Quality Aspects within the Holistic InfoSec Assurance Assessment Model (ISAAM)

After having analyzed all these aspects related to the concept of quality in a system or a process, we can state that *quality is a continuous improvement based on objective measurement*. The phrase *continuous improvement* recalls the PDCA approach described by ISO 27001:2005, an idea which was also brought up by Curkovic and Pagell (1999), stating that the improvement process is the underpinning of Total Quality Management, assimilating the quality process to success.

Considering all these aspects and in order to provide a quality system, an InfoSec program of a given organization should consider the following attributes:

- The existence within the Security Policy of the Security Dimensions and their respective Focus Areas ⇒ claiming the availability attribute of a quality process = evidence value 1;
- A designated person responsible for each Focus Area ⇒ claiming the responsibility attribute of a quality process = evidence value 2;
- Documented security Specific Factors to each Focus Area ⇒ claiming the implementation attribute of a quality process = evidence value 3;
- Monitored and audited Specific Factors of each Focus Area ⇒ claiming the effectiveness attribute of a quality process = evidence value 4.

It could be noticed that the evidence values go from a high-level perspective (evidence level 1, considering the Dimensions) to some more specific one (like the evidence value 4, considering the Specific Factors). As such, a Quality Average Weighting Level for each Dimension (QL_N) could be determined, based on the observation of the evidence:

$$QL_N \in \{1 \rightarrow 4\}.$$

Then, if needed, an overall InfoSec System Quality index (ISQ_i) could be calculated in order to measure the improvement level, or even to determine the current state of the quality related properties, where QL_N will represent the quality index of each dimension, and the number 16 will be the maximum value which could be attained, that is to say, an evidence value = 4 for all the four classes.

$$ISQ_i = \left(\sum_{Dimension\_1}^{4} QL_N \right) \div 16. \tag{5.1}$$

It clearly comes out from this formula that a further evidence value could not be reached if the previous one is not fully satisfied. This is because our model considers InfoSec from a holistic point of view, prioritizing thus transversal values rather than a silo approach. This is based on the idea that the InfoSec is a whole process rather than the sum of the different components.

5.3.3 The Requirements Side (Maturity Levels)

5.3.3.1 State of the Art Regarding the InfoSec Maturity Levels

Before the construction of our model, the authors performed a literature search on Maturity Models already in use in order to leverage the existing know-how in this field. The maturity models that were chosen to be consulted were selected in the way that the three main axes of evaluation, namely the InfoSec Structure, The

InfoSec System Quality, and the InfoSec Requirements, see Fig. 5.4, were represented. There are a lot of Maturity Models dedicated to the information security domain, but our analysis was focused on five of them, namely the CobiT Maturity Model (ISACA_&_ITGI 2007), Information Security Management Maturity Model (ISM3) (ISM 2007), the Common Criteria Assurance Levels (ISO-Std. ISO/IEC 15408:2005 2006b), the Systems Security Engineering Capability and Maturity Model (CMM) (ISSEA 2003; Lamnabhi 2008; van der Pijl et al. 1997), and the Maturity Model for IT operations (MITO) (Scheuing et al. 2000).

For the structural and organizational part, the relevant maturity models used within this chapter's framework are CobiT and ISM3. Both of these are focused on the management and governance dimension of information security. The first one concerns the internal control system, while ISM3 concerns the operation of the key processes of the Information Security Management System that are aligned with business objectives. CobiT's Maturity Model contains six levels and the ISM3 five levels. Considering the fact that a "nonexistent" level is not eligible for our evaluation model, the first level of CobiT was not retained (see the next section). Despite the similar focus of this two-maturity model on governance and security management, their point of view on security differs. CobiT is mostly focused on the way the internal controls are constructed and managed,[2] while ISM3 focuses on the existence or not of process metrics and the amount by which the security measures reduce security risks.

On the quality side, the CMM (both capability and maturity models) and MITO Maturity Model were studied. Regarding the CMM, both capability and maturity models were taken into account to analyze security process maturity. The capability and maturity levels of the process are very closely related since the capability model is related to the objectives in terms of security processes. The CMM maturity levels then take into account those objectives and use the content-related information to define the maturity levels. In that way the level 1 of the capability model, identified as *Performed*, corresponds to the *level 1* of the maturity model, where the processes are considered as being chaotic. The second level, identified as *Managed*, corresponds to the level 2 of the maturity model where the security process are planned and executed according to certain objectives. The third level, identified as *Defined,* corresponds to the *level 3* of the maturity model where the InfoSec human-related aspects are addressed through awareness activities. The fourth level of capability, identified as *quantitatively managed,* corresponds to the *level 4* of the maturity model where InfoSec controls measures are controlled and measured. Finally, the fifth level of capability identified as optimized includes the idea of continual improvement with respect to the maturity model. In the same way MITO identifies its five levels, following the same logic, moreover basing it on Maslow's hierarchy, which is: stochastic, repeatable, tracked, measured, and optimized.

On the technical and operational side, we based our reflection on the Common Criteria EAL levels. There are seven assurance levels defined within. Without analyzing them in detail, three groups of actions could be distinguished in order to pass

[2]CobiT's Maturity Levels: Non-existent, Initial/ad hoc, Repeatable but intuitive, Defined, Managed and Measurable, Optimized.

from one level to another. The first one considers the degree of knowledge regarding the security target, which constitutes the very first indicator bout the protection level to be provided. The second group is about the structure and the inherent features of the security mechanism, that is to say, the security system that will provide the necessary protection level to the target. The third group, which is about the upper levels, takes into consideration attributes such as the testing, review, and in-depth analysis of the security mechanisms.

5.3.3.2 The Holistic InfoSec Assurance Assessment Model (ISAAM) Maturity Level

Our Maturity Model is composed of five levels in order balance with the existing maturity models. The five levels defined by our IsAMM are: Fortuitous, Structured, Functional, Analyzed, and Effective. It is important to underline in this stage that the proposed IsAMM is strictly related to the evaluation model itself, using the same variables that are utilized inside. The IsAMM does not include a "nonexistent" level for two principal reasons. The first one is that because, as has been mentioned before, the Maturity level has to be coherent with the model. It was already specified that the proposed model could be applied when some prerequisites are achieved, see the section "A new Multidimensional InfoSec Assessment Framework." The second reason is that a "nonexistent" level would mean that the organization in question might have a risky attitude based on its own business objectives (profitability objectives). As such, there is no interest in applying such an evaluation model to fix a security level when no security is applied. For each level of maturity, considerations about evaluation model's components are defined in order to provide a homogeneous maturity model. Level 1 and Level 2 are characterized by the structure of the components of the security efforts, and from Level 3 the attention is mostly focused on the way the security activities are performed.

Level 1: Fortuitous Existence of at least one dimension but minimally the existence of the operational dimension is required. There are no identifiable and formal structures of "Focus Area" that really exist, although some isolated "Specific Factors" may be perceived. This means that the existing security activities are instinctively performed within the organization. These security activities take the shape of some security measures applied as a result of mandatory requirements or basic needs for protection. Consequently InfoSec is not at the "managed process" level; and as such, no quality specification can be made. More generally speaking, an organization which finds itself in a Level 1 position does not display any proactive behavior regarding the protection of its assets.

Level 2: Structured There is a clear and formal existence of all four InfoSec dimensions, as shown within the evaluation model. The formal existence of the security dimensions could be claimed if some precise objectives regarding the dimensions are specified. Consequently, the elements of the "Focus Area" mostly exist, as

demanded by the best practices, baseline approaches, and described by the evaluation model. To each focus area, a minimum of two "Specific Factors" are dedicated, more often the "Procedure/Controls" and "Resource" attributes could be assigned. The quality level reaches the second stage, that is to say, for each dimension, a responsible person could be found. More generally speaking, an organization which finds itself in a Level 2 position may claim the existence of a security program running inside it.

Level 3: Functional There is a complete InfoSec architecture as described by the evaluation model. That means that a clear structure exists in terms of the Organizational, Operational, Human, and Legal dimensions. To each one of the aforementioned criteria, the major parts of the Focus Area are identified. This brings the organization to an accountability stage, where the organization has a clear understanding of the risks it may face and the related issues. Regarding the "Specific factors," those related to the "Procedure/Controls" and "Resources" obligatorily should exist, because of the accountability level claimed by the functional level. As mentioned before, from this level InfoSec is not only considered as an ensemble of countermeasures but as an operational system composed of some defined and complete processes and procedures. For this reason, the average quality weighting level equals three or four, which means that Specific Factors are minimally documented and monitored.

Level 4: Analyzable Organizations judged as having reached the fourth level of this maturity model should represent a completed architecture in terms of Dimension and Focus Area. Based on the construction of the evaluation model, it could be claimed that the organizations that have reached the fourth level apply the major part of the requirements expected by the well-known InfoSec standards, best practices, etc. In that way it could be stated that the fourth-level organization possesses a discernible managed InfoSec System. Based on this the average quality, weighting level clearly equals the fourth level.

Level 5: Effective This is the ultimate stage the organizational InfoSec system could reach, incorporating the continual improvement reflex. As such, a completed architecture in terms of Dimensions and Focus Area fully exists. Regarding the Specific Factors, all three attributes could be discerned, namely the operational measures, appropriate human resources, and the appropriate procedures, see the "Expected outputs of the ISAA Evaluation Model" section. The average quality weighting level attains the fifth level meaning that a measurement system is run inside the organization. Reviews and audits are regularly performed to fix the current situation and future ones in InfoSec terms, fulfilling in that way the existing gaps. A continual improvement procedure is imputed to each Focus Area.

5.3.4 Expected Outputs of the ISAA Evaluation Model

The model proposed within the context of this chapter aims to be a semiformal one, which means that a natural language is used. This is based on a specific method imposing a rigorous structure of the processes. In a first stage, we chose to structure the InfoSec Program according to the Common Criteria modeling concept. This is because our objective is to evaluate a function, that is to say, Information Security, including the aforementioned four principal dimensions. The Information Security Level of a given organization will derive from the performance quality of each one of these dimensions. In that way the objective to provide a holistic evaluation will be fulfilled.

As has been mentioned above, the assurance argument regarding a system under evaluation could be derived from a nested structure. For that reason, the Information Security program is structured in a nested manner including Dimensions, Focus Area (FAs) for each Dimension, and Specific Factors (SFs) for each Focus Area. As we have seen above, the structural assurance is one of the conditions regarding the trust in a given system, see trust-related section. This means that in order to gain assurance over the system, a formal structure, dissecting all the components of the system, is needed. This has to be done from a general view of the system to the most specific evidence components, which are the roots-of-trust corresponding to the different attributes within our model.

From a conceptual point of view, InfoSec Program is the variable under evaluation. The different levels resulting from the evaluation will define the protection level of the organizational values. This variable has to be evaluated and assessed holistically. For that, the four dimensions of the variable are identified and analyzed.

Each of these dimensions includes different tools, namely the FAs which are sets of activities responding to the same objective within the security program. Then a set of components for each FA is identified representing the mechanism that permits the achievement of the security objective of the family, according to the classification made by the Common Criteria, see Fig. 5.5. As stated in McKnight and Chervany (2000), trust is the belief that proper impersonal structures are in place in order to enable one to anticipate a successful future endeavor.

A similar structure based on the Security Dimensions was adapted from the Organization for Economic Cooperation and Development guide, which included some guidelines for security information systems and networks (OECD 2002) and was published in July 2002. Within this publication, Information Security is considered in a very broad sense, and elements enumerated inside the guide consider the Information Security realm in a holistic way. This means that under the assumption that the security level will be as strong as the weakest link, the idea of evaluating all security components (technical or not) contributes to minimize the likelihood of the existence of the weakest link causing the breach and leading to harm.

Following the model's structure presented in Fig. 5.5, for each Dimension, the following FAs are identified:

- **Organizational dimension**, divided into two subdimensions, Management and Governance, is made up of the following FAs: Strategy, Roles and Responsibil-

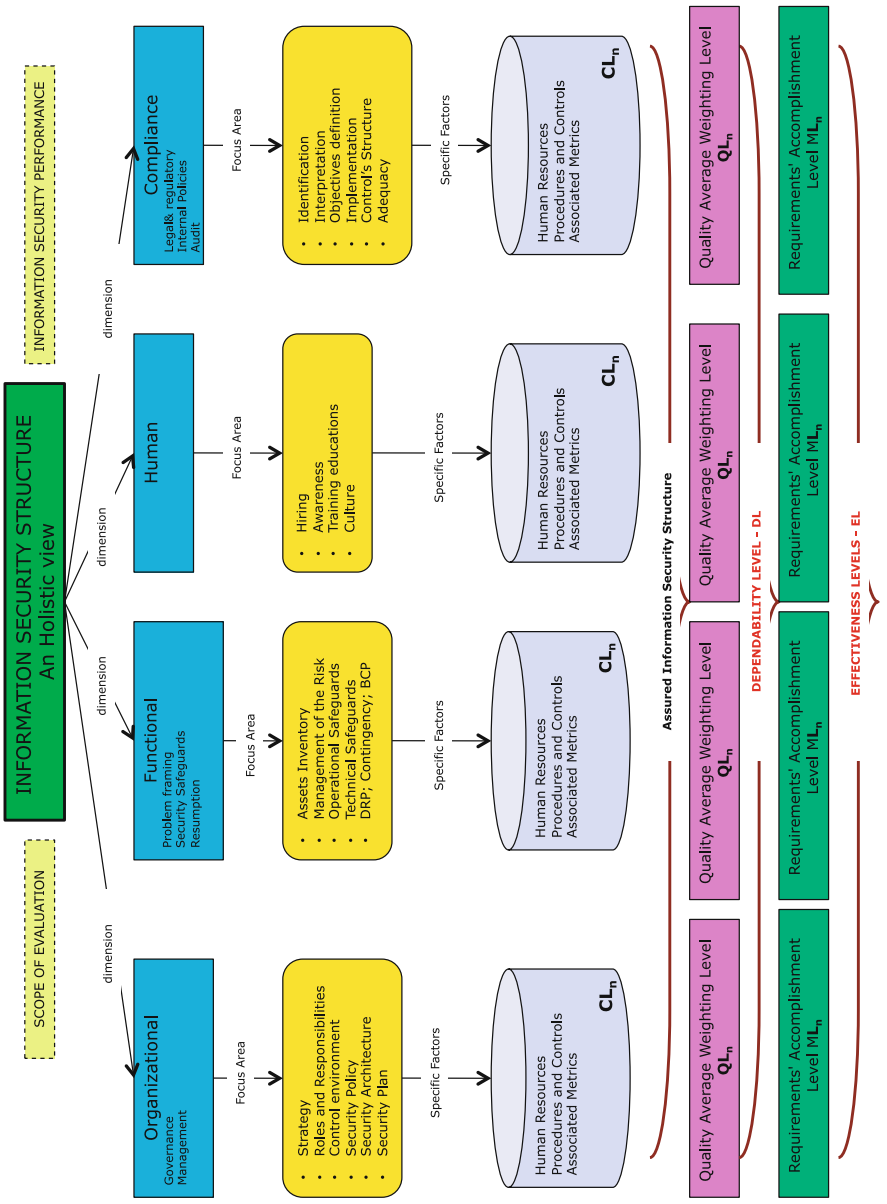


Fig. 5.5 The ISAA evaluation model

ities, Control Environment, InfoSec Policy, InfoSec Architecture, and InfoSec Plan.

- **Operational dimension**, divided into three subdimensions, Problem Framing, Security safeguards, and Resumption, is made up of the following FAs: Assets

Inventory, Risk Identification, Risk Assessment, Operational Safeguards, Technical Safeguards, Business Continuity and Disaster Recovery Planning.

- **Human dimension** is made up of the following FAs: Hiring and staffing, Awareness, Training and Education, and the overall Security Culture.
- **Legal dimension**, divided into three subdimensions, Legal & Regulatory, Internal Policies and Audit, is made up of the following FAs: Identification, Interpretation, Objectives definition, Implementation, structure of controls, Adequacy.

Each one of the issues contained within the FAs is resolved by some SFs (roots-of-trust) that within the model are gathered into three groups: InfoSec Measure & Activities,[3] InfoSec Procedures,[4] and InfoSec Human Resources.[5] This "assured structure" made up of these three elements allows the calculation of the InfoSec Completeness Level index that aims to evaluate the assurance level of the InfoSec program's structure:

$$IsCL = \sum_{i=1}^{n} \{SF_{s\text{attributes}} / N_{\text{FAs}}\} \div 3. \tag{5.2}$$

This model allows the calculation of two main holistic InfoSec Assurance indices, namely the *Dependability Level* (D_L) and *the Effectiveness Level* (E_L).

The first index includes and captures two main aspects; trustworthiness on the InfoSec Program structure, by including the appropriate index calculated before and the quality assurance by including as a weighting factor the quality level QL_N, where $QL_N \in \{1 \rightarrow 4\}$, see "The Security Quality" section above.

$$D_L = \sum_{i=1}^{n} (IsC_{Li} * Q_{Li}). \tag{5.3}$$

The second index, concerning the Infosec Program's Effectiveness, is calculated as shown:

$$E_L = \sum_{i=1}^{n} D_{Li} / M_{Li}. \tag{5.4}$$

In fact, based on the definition of the effectiveness, our index incorporates two concepts, the dependability and the maturity. The dependability represents the state of the overall InfoSec Program on time "t", while the Maturity Level as defined by the model, see "The Requirements side (Maturity Levels)," section, serves as the Requirements reference point and consequently as objectives to be accomplished.

[3] Including technical, operational, and organizational measures.

[4] Including frameworks, verification, and reporting procedures.

[5] Including the dedicated staff, responsible persons, and managements bodies.

5.4 Conclusion

The aim of this chapter, based on our current research, was to present a conceptual model describing the way to holistically assess the Information Security Assurance Posture. The model is inspired by some well-known security standards. The idea was to map the engineering security standards within the nonengineering assessment models of the Information Security, in order to formalize the way that Information Security is evaluated. The model permits the evaluation of the InfoSec Program by accomplishing the "*assurance requirement*" by taking into account two main characteristics:

- *Effectiveness*, which means that the system under evaluation is doing the right thing, and
- *Efficiency*, which means that the system under evaluation is doing things right.

The evaluation model proposed within this chapter could be performed internally and externally. As has been emphasized several times, the model prioritizes a transversal approach and implies a deep understanding of the organization through the knowledge of organizational requirements. Based on this, the size of the organization will not strongly influence the assurance level, except in its first stage which deals with the InfoSec assurance structure. But even in the case of the assurance structure, the model assumes that a certain security baseline should be provided. As such, and independently of the organization's size, all the four dimensions should exist and should be operated, as a minimum of attributes in order that an InfoSec program or system might be claimed.

Acknowledgements The authors would like to thank Mr. David Simms, postgraduate student at HEC, University of Lausanne, for his valuable assistance in proof-reading this chapter.

References

B. Barafort, J.-P. Humbert, and S. Poggi, "Information security management and ISO/IEC 15504: the link opportunity between security and quality," in *Proceedings of The Sixth International Software Process Improvement and Capability Determination (SPICE) Conference*, Luxembourg, 2006.

K. Beznosov and P. Kruchten, "Towards agile security assurance," in *Proceedings of the 2004 workshop on New security paradigms*, Nova Scotia, Canada, 2004, pp. 47–54.

S. Butler, "Security attribute evaluation method: a cost-benefit approach," in *Proceedings of the 24th International Conference on Software Engineering*, Orlando: ACM, 2002.

T. Chamfrault and C. Durand, *ITIL et la Gestion des Services—Méthodes, Mise en Oeuvre et Bonnes Pratiques*. Paris: Dunod, 2006.

S. Curkovic and M. Pagell, "A critical examination of the ability of ISO 9000 certification to lead to a competitive advantage," *Journal of Quality Management*, vol. 4 (1), pp. 51–67, 1999.

W. DeLone and E. McLean, "Information systems success: the quest for the dependent variable," *Information Systems Research*, vol. 3 (1), pp. 60–95, 1992.

W. DeLone and E. McLean, "The DeLone and McLean model of information system success: a ten-year update," *Journal of Management Information Systems*, vol. 19 (4), pp. 9–30, 2003.

Department_of_Defense_(USA), *Department of Defense Trusted Computer System Evaluation Criteria (TCSEC)*, Washington, USA, 1985.

S. Feldman, "Quality assurance: much more than testing," *ACM Queue*, vol. 3 (1), pp. 26–29, 2005.

F. Gallegos, S. Senft, D. Manson, and C. Gonzales, *Information Technology Control and Audit*. Washington: Auerbach, 2004.

S. G. Herrero, M. A. M. Saldana, M. A. M. d. Campo, and D. Ritzel, "From the traditional concept of safety management to safety integrated with quality" *Journal of Safety Research*, vol. 33 (1), pp. 1–20, 2002.

ISACA_&_ITGI, *Control Objectives for Information and related Technology (COBIT)*, Information Systems Audit and Control Association and IT Governance Institute, 2007. [Online] Available at http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=7981

ISF-std, *The Standard of Good Practice for Information Security*, Information Security Forum, 2007.

ISM^3, "Information Security Management Maturity Model," ISM3 Consortium, Madrid, Spain 2007. [Online] Available at http://www.ism3.com/index.php?option=com_docman&task=cat_view&gid=1&Itemid=9

ISO-Std. ISO/IEC 27002:2005, *Information technology—Security techniques—Code of practice for information security management*, International Organization for Standardization (ISO), Switzerland, 2005.

ISO-Std. ISO/IEC 27001:2005 (E), *Information Technology—Security Techniques—Information Security Management Systems—Requirements*, International Organization for Standardization (ISO), Switzerland, 2005.

ISO-Std. 9001:2000, *Quality Management Systems—Requirements*, International Organization for Standardization (ISO), Switzerland, 2000.

ISO-Std. 9000:2005, *Quality Management Systems—Fundamentals and Vocabulary*, International Organization for Standardization (ISO), Switzerland, 2005.

ISO-Std. ISO/IEC 15408:2005, *Information technology—Security techniques—Evaluation criteria for IT security, Part 1: Introduction and general model*, International Organization for Standardization (ISO), Switzerland, 2006a.

ISO-Std. ISO/IEC 15408:2005, *Information technology—Security techniques—Evaluation criteria for IT security, Part 2: Security functional components*, International Organization for Standardization (ISO), Switzerland, 2006b.

ISO-Std. ISO/IEC 15408:2005, *Information technology—Security techniques—Evaluation criteria for IT security, Part 3: Security assurance components*, International Organization for Standardization (ISO), Switzerland, 2006c.

ISSEA. *Systems Security Engineering Capability Maturity Model (SSE-CMM)*, International Systems Security Engineering Association (ISSEA), 2003.

G. F. Jelen and J. R. Williams, "A practical approach to measuring assurance," in *Proceedings of 14th Annual Computer Security Applications Conference*, 1998, pp. 333–343.

S. Karapetrovic and W. Willborn, "Quality assurance and effectiveness of audit systems," *International Journal of Quality & Reliability Management*, vol. 17 (6), pp. 679–703, 2000.

I. Koskosas, "Goal Setting and Trust in a Security Management Context," *Information Security Journal: A Global Perspective*, vol. 17 (3), pp. 151–161, 2008.

M. Lamnabhi, *Evaluer avec CMMI—Etape par Etape*, Paris: AFNOR Editions, 2008.

Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang, "AIMQ: a methodology for information quality assessment," *Information & Management*, vol. 40 (2), pp. 133–146, 2002.

M. Ludwig-Becker, "Quality management principles as top team performance practices. ISO 9000 re-interpreted," *Team Performance and Management*, vol. 5 (7), pp. 207–211, 1999.

S. Maynard and A. B. Ruighaver, "What makes a good information security policy: a preliminary framework for evaluating security policy quality," in *Proceedings of the 5th Annual Security Conference*, Las Vegas, Nevada, USA, 2006.

D. H. McKnight and N. L. Chervany, "What is trust? A conceptual analysis and an interdisciplinary model," in *Proceedings of the 2000 Americas Conference on Information Systems*, California, USA, 2000, pp. 827–833.

R. Mercuri, "Standards insecurity," *Communications of the ACM*, vol. 46 (12), pp. 21–25, 2003.

M. Merkow and J. Breithaupt, *Computer Security Assurance Using the Common Criteria*. New York: Thomson Delmar Learning, 2005.

OECD, "OECD Guidelines for the Security of Information Systems and Networks Towards a Culture of Security," Organisation for Economic Co-operation and Development, Paris, 2002. [Online] Available at http://www.oecd.org/document/42/0,3343,en_21571361_36139259_15582250_1_1_1_1,00.html

Office_for_Official_Publications_of_the_European_Communities, *Information Technology Security Evaluation Criteria (ITSEC)*, Luxembourg, 1991.

G. J. van der Pijl, G. J. P. Swinkels, and J. G. Verrijdt, "ISO 9000 versus CMM: standardization and certification of IS development," *Information & Management*, vol. 32 (6), pp. 267–274, 1997.

P. Ragozzino, "IS quality—what is it?," *Journal of Systems Management*, vol. 41 (11), pp. 15–16, 1990.

A. Q. Scheuing, K. Frühauf, and W. Schwarz, "Maturity model for IT operations (MITO)," in *Proceeding of the 2ndWorld Congress on Software Quality*, Yokahoma, Japan, 2000.

J. Slay, "IS Security, trust and culture: a theoretical framework for managing IS security in multicultural settings," *Campus-Wide Information Systems*, vol. 20 (3), pp. 98–104, 2003.

I. Tashi and S. Ghernaouti-Hélie, "La certification comme référentiel de classification da la sécurité," in *Proceedings of the AFME Colloque—Association Francophone de Management Electronique*, Montréal, Canada, 2006, CD-ROM, Alphabetical list of the communications.

I. Tashi and S. Ghernaouti-Hélie, "ISO security standards as a leverage on IT Security Management," in *Proceedings of 13th Americas Conference on Information Systems (AMCIS)*, Colorado, USA, 2007, Paper 63.

I. Tashi and S. Ghernaouti-Hélie, "Regulatory Compliance and Information Security Assurance," in *The First International Workshop on Global Information Security for an Inclusive Information Society (GloSec), The International Dependability Conference (ARES)*, Fukuoka, Japan, 2009, pp. 670–674.

U.S.NRC, "Quality assurance criteria for nuclear power plants and fuel reprocessing plants," in *RC Regulations Title 10, Code of Federal Regulations: Requirements Binding on all Persons and Organizations Who Receive a License from NRC to Use Nuclear Materials or Operate Nuclear Facilities*, United.States_Nuclear_Regulatory_Commission, Ed., 2000.

I. Verbauwhede and P. Schaumont, "Design methods for security and trust," in *The Proceedings of the Conference on Design, Automation and Test in Europe*, Nice, France, 2007, pp. 672–677.

J. Williams and G. Jelen, "A Framework for Reasoning about Assurance," NSA, 1998. [Online]. Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.5581&rep=rep1&type=pdf

J. Williams, J. Sachs, D. Landoll, and D. Carpenter, "Assurance is an N-space, where N is hopefully small," in *Proceedings of International Invitational Workshop on Developmental Assurance*, 1994.

J. R. Williams and G. F. Jelen, "A practical approach to improving and communicating assurance," in *Proceedings of the 11 th Annual Canadian Information Technology Security Symposium*, 1999.

Z. Yan and S. Holtmanns, "Trust modeling and management: from social trust to digital trust," in *Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions*, IGI Global, 2007.

Chapter 6
Risk-Aware Business Process Management—Establishing the Link Between Business and Security

Stefan Jakoubi, Simon Tjoa, Sigrun Goluch, and Gerhard Kitzler

Summary Companies face the challenge to effectively and efficiently perform their business processes and to guarantee their continuous operation. To meet the economic requirements, companies predominantly apply business process management concepts. The substantial consideration of robustness and continuity of operations is performed in other domains such as risk or business continuity management. Applying these domains separately, analysis results may significantly differ as valuations from an economic and risk point of view may lead to deviating improvement recommendations. Observing developments in the past years, one can see that regulative bodies, the industry, and the research community laid a special focus on the tighter integration of business process and risk management. Consequently, the integrated consideration of economic, risk, and security aspects when analyzing and designing business processes delivers enormous value to achieve these requirements.

In this chapter, we present an survey about selected scientific approaches tackling the challenge of integrating economic and risk aspects. Furthermore, we present a methodology enabling the risk-aware modeling and simulation of business processes.

S. Jakoubi (✉) · S. Goluch · G. Kitzler
Secure Business Austria, 1040 Vienna, Austria
e-mail: sjakoubi@sba-research.org

S. Goluch
e-mail: sgoluch@sba-research.org

G. Kitzler
e-mail: gkitzler@sba-research.org

S. Tjoa
St. Poelten University of Applied Sciences, 3100 St. Poelten, Austria
e-mail: simon.tjoa@fhstp.ac.at

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_6, © Springer Science+Business Media, LLC 2010

6.1 Introduction

Maximizing revenues has always been and will always be the outmost objective of profit oriented companies. Business process management assured within the last years its position as predominant player in modeling and simulating a company's business processes providing significant decision support in optimizing workflows and resource utilizations. Thus, it is no surprise that Gartner outlines in its CIO report 2009 (Gartner Inc. 2009) that the improvement of business processes is number one priority. Let us take a closer look at the statement "improvement of business processes." It is obvious that for profit maximizing ambitions, the economic effectiveness and efficiency of business processes has to be optimized. The reduction of execution and waiting times, more efficient process activity structures, and resource allocations are only a few examples how to improve the executed business processes from an economic point of view. At the same time, one must not forget to in-depth consider requirements from business on its processes such as confidentiality, integrity, and availability in order to mention the most popular security goals. The best possible optimized business process is worthless if it cannot be executed, for example, in the case of a complete data center outage. Serious legal implications would arise if highly sensible health data is disclosed to unauthorized entities. A company can hardly be satisfied if, for instance, the car manufacturing process is accelerated for ten percent but a resulting recall initiative annihilates this improvement and furthermore requires significant additional budget for taking reputation rehabilitation actions. There exist diverse classifications of these threats (National Institute of Standards and Technology 2002; BSI 2004; International Organization for Standardization 2004) ranging from accidents (e.g., unavailability of ICT resources or the absence of strategic personnel) to natural catastrophes (e.g., earthquakes) and to deliberate acts (e.g., sabotage or theft). Risk management has been the major player addressing these issues. In the past years it got significant support through the evolvement and acceptance of further domains such as incident, disaster recovery, and business continuity management (National Institute of Standards and Technology 2004; British Standard Institute 2006, 2007; International Organization for Standardization 2008). The European Network and Information Security Agency (ENISA) states that "it is very difficult to isolate all the disciplines related to planning for and recovering from an incident which threatens an organization either from an internal or external source. All the disciplines are closely related and there are areas of cross-over…" (European Network and Information Security Agency 2008). However, diverse associations and regulative bodies emphasized the importance of seriously tackling risks while improving business performance which became manifest in exemplarily the Sarbanes Oxley Act (One Hundred Seventh Congress of the United States of America 2002) or the 8th audit directive of the European Union (European Commission 2010). Searching in relevant libraries, one can furthermore observe that over the last years also the scientific community increased its research efforts in trying to integrate risk and economic business aspects. As mentioned above, business process modeling and simulation is the adequate technique to

support the economic analysis of a company's business processes. Simultaneously, there are several research results regarding the integration of risk aspects and security requirements into business process analyses. However, these approaches do not focus on modeling characteristics that are required for performing (risk-aware) business process simulations. As a consequence, business process simulations support economic analyses and optimizations but neglect the consideration of security and business continuity requirements.

The major objective of this book chapter is to address these shortcomings. Therefore, we, on the one hand, provide selected related research and, on the other hand, present our approach for risk-aware business process management. The term risk-aware business process management is understood as the integration of a risk perspective into business process management. The rest of this chapter is organized as follows. Section 6.2 gives an overview about selected related research. Section 6.3 provides information about required steps to perform risk-aware business process management. In Sect. 6.4, we introduce our proposed reference model. In Sect. 6.5, we outline the business case for applying our approach and give examples for application scenarios. We conclude this chapter in Sect. 6.6 and give an outlook on future research challenges.

6.2 Related Work

In this section, we give a brief overview about selected research results aiming at the incorporation of risk aspects into business process modeling and analyses. For detailed information on the included approaches, we kindly refer to the according references of the provided related work.

Sackmann extends current risk management methods with a business process-oriented view leading to an IT risk reference model (Fig. 6.1), which builds the bridge between the economic and more technical layers including vulnerabilities (Sackmann 2008; Sackmann et al. 2009). The introduced model consists of four interconnected layers: (1) Business process layer: A business process consists of activities and sub-processes. To quantify IT risks, it is necessary that the monetary value of the process for the company can be calculated. (2) IT applications/IT infrastructure layer: this layer comprises all required IT applications and underlying infrastructure components. (3) Vulnerabilities layer: the layer includes "...all vulnerabilities that exist in the components..." (Sackmann 2008) of the IT applications/IT infrastructure layer. (4) Threats layer: this layer comprises all threats that can result in IT risks. Ideally, the occurrence probability should be determined. This reference model "serves as foundation for formal modeling of the relations between causes of IT risks and their effects on business processes or a company's returns" (Sackmann 2008). For expressing these relations (i.e., the searched cause-effect relations), a matrix-based description is used.

CORAS (Braber et al. 2007) is a method for conducting security risk analysis, which is abbreviated to "security analysis." CORAS provides a customized language for threat and risk modeling and comes with detailed guidelines explaining

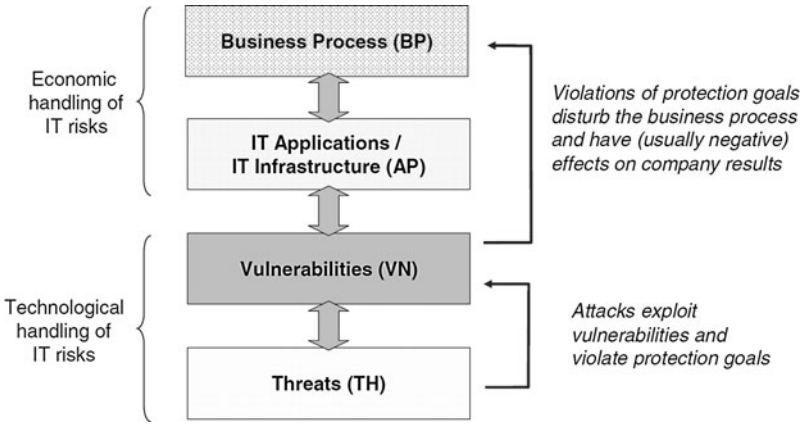


Fig. 6.1 IT risk reference model (Sackmann 2008)

how the language should be used to capture and model relevant information during the various stages of the security analysis. The Unified Modeling Language (UML) is used to model the target of the analysis. For documenting intermediate results and for presenting the overall conclusions, special CORAS diagrams which are inspired by UML are used. The CORAS approach comprises the succeeding seven steps. (1) Introductory meeting: Information gathering is performed through an introductory meeting. The representatives of the client present their goals of the analysis and the target to be analyzed. (2) High-level analysis: Separate meetings with the representatives where the analysts present their understanding of what they learned at the first meeting and from studying documentation which have been provided by the client. The meeting includes a first high-level security analysis where threats, vulnerabilities, threat scenarios, and unwanted incidents are identified. This input is used to direct and scope the further detailed analysis. (3) Approval: Refining the description of the target to be analyzed and identifying all assumptions and other preconditions being made. (4) Risk identification: Through a workshop with experienced people as many potential unwanted incidents, threats, vulnerabilities, and threat scenarios as possible are identified. (5) Risk estimation: Through a workshop estimates on consequences and likelihoods of unwanted incidents are identified. (6) Risk evaluation: Presenting the client the first overall risk picture. This typically triggers adjustments and corrections. (7) Risk treatment: Through a workshop treatment and cost/benefit issues are identified.

Karagiannis et al. (2007) present in their work a business process-oriented approach to support Sarbanes Oxley Act (SOX) compliance efforts of organizations. The authors propose a six-step approach supported through the ADONIS® platform. Furthermore, they extended the ADONIS® standard modeling language in order to meet the requirements demanded by SOX and COSO. The six-step framework consists of the following phases: (1) Business Process Acquisition: Business processes serve as the foundation of the approach and are therefore acquired within the first step. (2) Risk Assessment and Scoping: In a second step, SOX-related risks (in-

cluding likelihood and impact) are identified and modeled. The relation between the risk and the concerned business process is also addressed. Moreover, controls are documented using a control model. (3) Design Effectiveness: This stage "...deals with the revision of internal controls, intended to balance risk and control costs..." (Karagiannis et al. 2007). (4) Operating Effectiveness: The aim of this step is the evaluation of the effectiveness of the current internal control set during operations. The authors propose self assessments, internal audit reviews, or testing procedures as possible sources to determine the effectiveness. (5) Internal Management Review: This stage assesses predefined goals of the company against the test results of the previous steps to determine if the company is SOX-compliant. (6) Auditor's Final Review: Within the last step "...the external auditor receives financial reports along with internal management review reports..." (Karagiannis et al. 2007). The evaluation of this approach was performed at an US insurance company covering 180 business processes. Further details about the approach and the evaluation can be found at Karagiannis et al. (2007).

AURUM is a framework for automated information security risk management (Ekelhart et al. 2009a, 2009b; Fenz et al. 2009). As basis for their research, the authors identify the following questions which have to be addressed by organizations: (1) What are potential threats for my organization? (2) How probable are these threats? (3) Which vulnerabilities could be exploited by such threats? (4) Which controls are required to most effectively mitigate these vulnerabilities? (5) What is the potential impact of a particular threat? (6) What is the value of security investments?, and finally (7) In which security solutions is it worth investing? The research focuses on developing concepts to meet these demands of the information security risk management (ISRM) community with the aim to support risk managers in making efficient security decisions. The detailed specification of the developed concepts introduces new automated risk management approaches on a conceptual level and poses as template for tool implementations. Figure 6.2 shows how the main ISRM-phases are supported. The purpose of the entire framework is to support investment decision makers in interactively selecting efficient security solutions. The ISRM process starts at the business process importance phase, where importance values are assigned for each required asset. Based on business process models and an overall importance value for each, asset importance values are automatically calculated. In the inventory phase, the organizations has to define (i) their assets, (ii) the acceptable risk level of the defined assets, (iii) the organization-wide importance of the defined assets, and (iv) the attacker profile in terms of motivation and capability. To store and interrelate this information with general information security domain knowledge, the authors use a security ontology. In the threat probability phase the developed Bayesian threat probability determination extracts knowledge regarding threats, threat a priori probabilities, vulnerabilities, existing and potential control implementations, attacker profiles, and the assets of the organization from the security ontology and establishes a Bayesian network capable of calculating threat probabilities based on the aforementioned input information. In the risk determination phase relevant threat probabilities are merged with the importance information regarding the considered asset. In the control identification and evaluation phase existing and potential control implementations, their effectiveness, initial and running

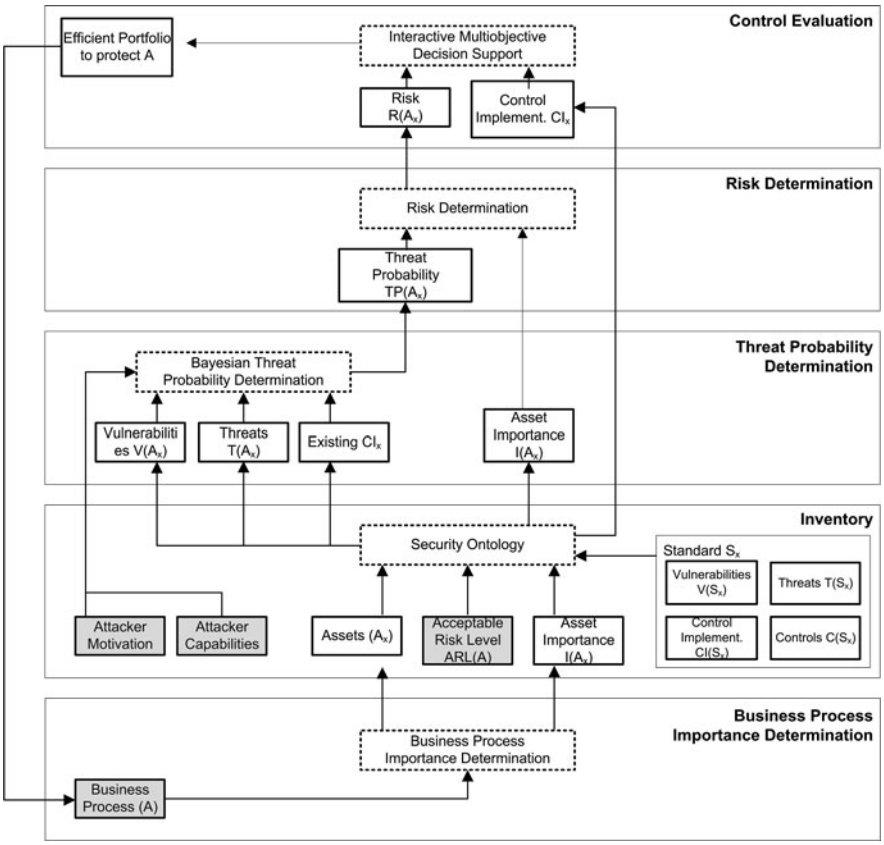


Fig. 6.2 The AURUM process

costs are extracted from the security ontology to support the developed interactive multicriteria decision support. Information regarding the relevance of existing and potential control implementations is extracted from the Bayesian threat probability model. Using the extracted data as input for the developed multicriteria decision support methodology, a solution concept is provided for two fundamental ISRM questions: (i) Which IT security solutions can generally be used to mitigate the risk to an acceptable level?, and (ii) Which IT security solutions should be used to mitigate the risk cost-efficient to an acceptable level?

Modeling security requirements in business processes is also the goal of an extension of UML 2.0 by Rodríguez et al. (2006). According to the authors, this is essential since software developers derive necessary requirements for software design and implementation from business processes (Rodríguez et al. 2006). This early design of security requirements shall (1) use the (at least high-level) security knowledge of business analysts concerning business process security while initially modeling the processes and (2) reduce potential costs avoiding the additional implementation

of business processes' security after the business processes have been implemented. The proposed extension makes use of activity diagrams to allow the definition of business processes security requirements (Fig. 6.3).

Zur Muehlen and Rosemann identify risk as an inherent property of every business process (zur Muehlen and Rosemann 2005). Therefore they propose to counteract the trend of considering risk only from a project management viewpoint and to tackle the topic of risk management in the context of business process management. They consequently introduce a taxonomy (Fig. 6.4) including process-related risks and their appliances concerning the analysis and documentation of business processes. Additionally, they propose a taxonomy for business processes including five clusters (goals, structure, information technology, data, and organization) and two distinguished lifecycles (build-time and run-time), enabling the classification of both, errors and risks. To capture risks in the context of business processes, the authors introduce four interrelated model types:

1. The Risk Structure Model provides information regarding the relationship between risks.
2. The Risk Goal Model represents a risks/goals matrix.
3. The Risk State Model captures the dynamic aspects of risks and consists of the different object types: risk, consequence, and connectors.
4. Event-driven Process Chains (EPCs) are extended to consider risks, enabling the assignment of risks to individual steps in the specific process.

The need for a holistic business view on risk management is addressed by Neiger et al. (2006). Accordingly, value-focused process engineering, which creates links between business processes and business objectives at the operational and strategic levels, is utilized. This value-focused process engineering approach is applied to risk management models, resulting in a risk-oriented process management view. The overall model consists of four steps:

1. To identify relevant process risks, business objectives are decomposed, while each process activity is examined in order to identify further relevant risks.
2. To identify risks and to determine related processes, value-focused approaches are used.
3. To identify the best process structure to meet the business objectives, process configurations are suggested.
4. To enable the selection of an optimal process configuration, alternative configurations and their corresponding results that meet the identified risk minimization objectives are finally compared.

Focusing on business process availability, Milanovic et al. (2008) present a framework for modeling availability considering services, underlying ICT infrastructure and human resources. To model these relations, the authors adapt a service-enabled architecture (Fig. 6.5). Moreover, a fault-model with two failure modes (Temporal/Value) is used, thus enabling an analytical assessment procedure:

1. Define the business process following a process modeling language.
2. Refine activities by modeling atomic services.

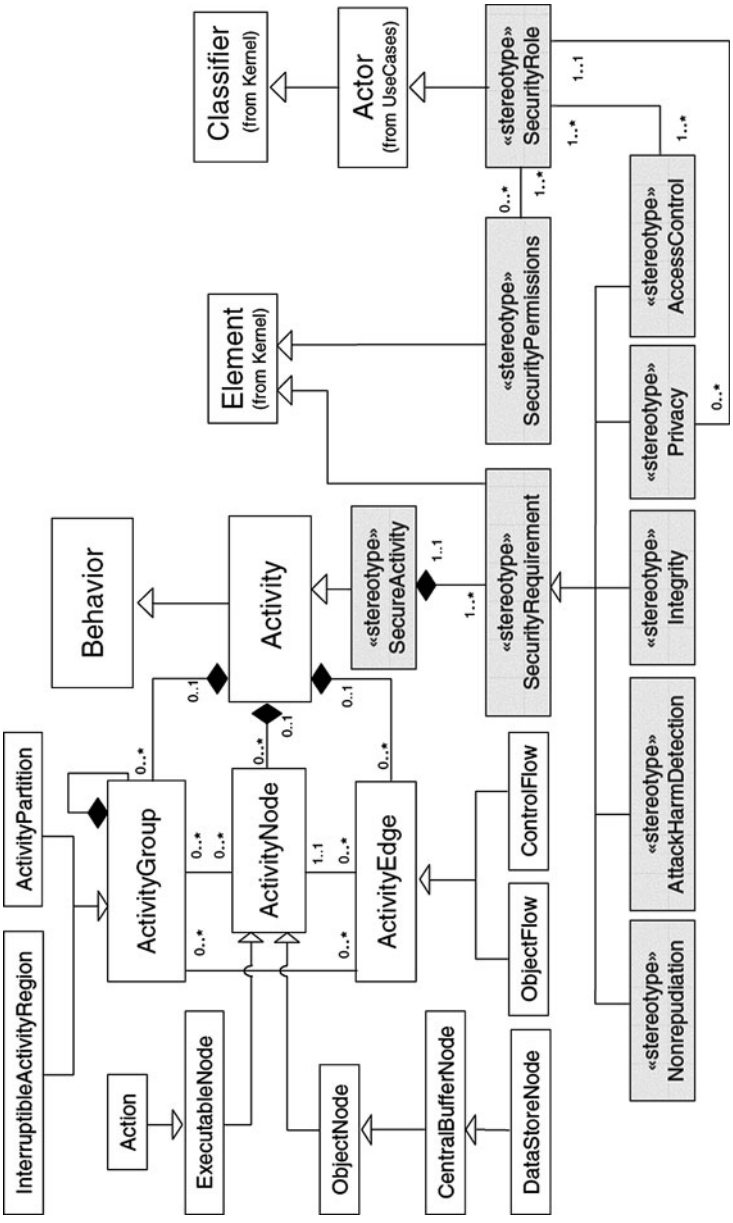


Fig. 6.3 Extending the UML 2.0 metamodel with security stereotypes (Rodríguez et al. 2006)

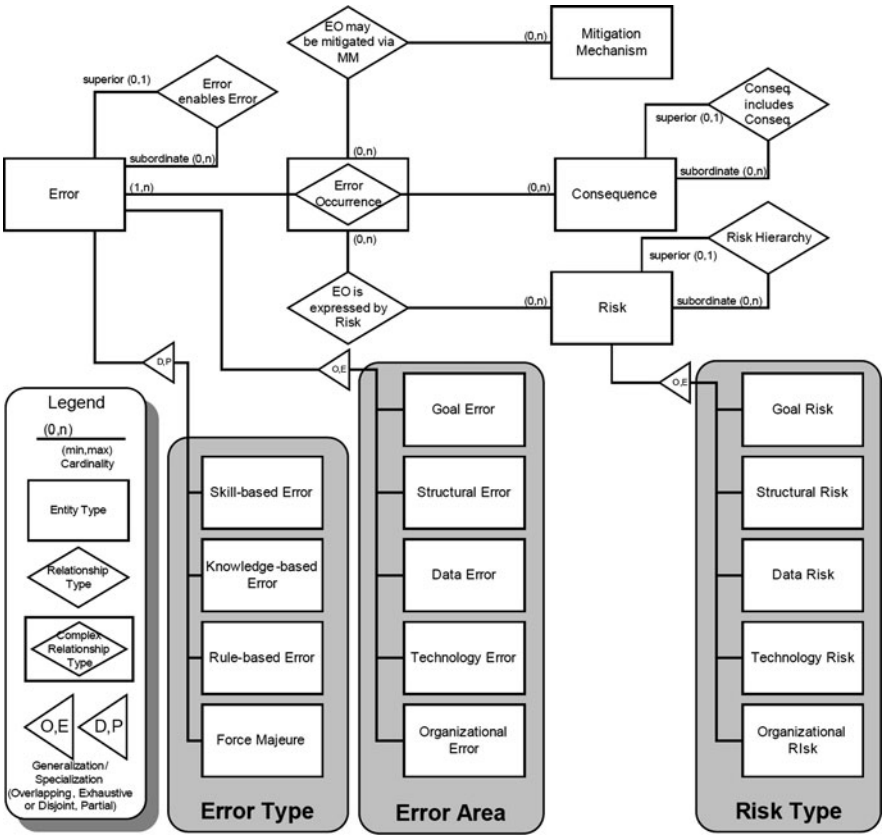


Fig. 6.4 Risk taxonomy (zur Muehlen and Rosemann 2005)

3. Create an infrastructure graph.
4. Map services to infrastructure components. Transform paths for service executions into Boolean expressions.
5. Map business processes to atomic services. The functional dependency between business process, service and ICT-layer availability is the result.
6. Transform the Boolean expressions into reliability block diagrams/fault trees to calculate steady-state availability.
7. Calculate the availability of business process and services by solving/simulating the model generated within the abovementioned steps.

Regarding the compliance of business processes, Weber et al. (2008) propose an approach to validate whether the states reached by a process are compliant with a set of constraints or not. This enables compliance checking of a new or altered process against a given constraints base and of the process repository against a different or changed constraints base (Fig. 6.6). The authors formalize and utilize a class of compliance rules and annotated process models respectively.

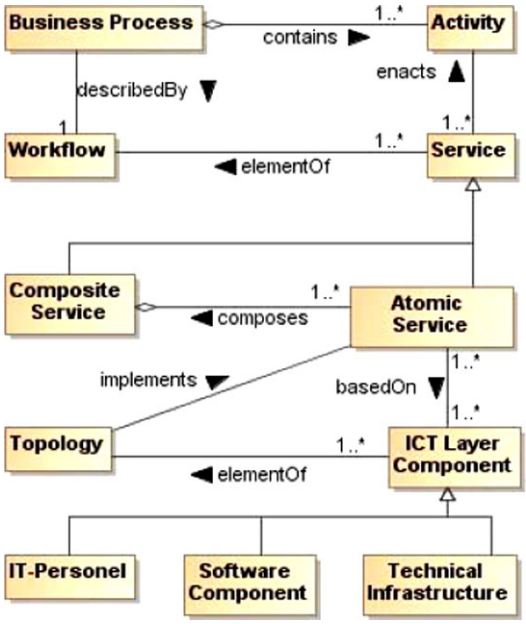


Fig. 6.5 Service-enabled architecture (Milanovic et al. 2008)

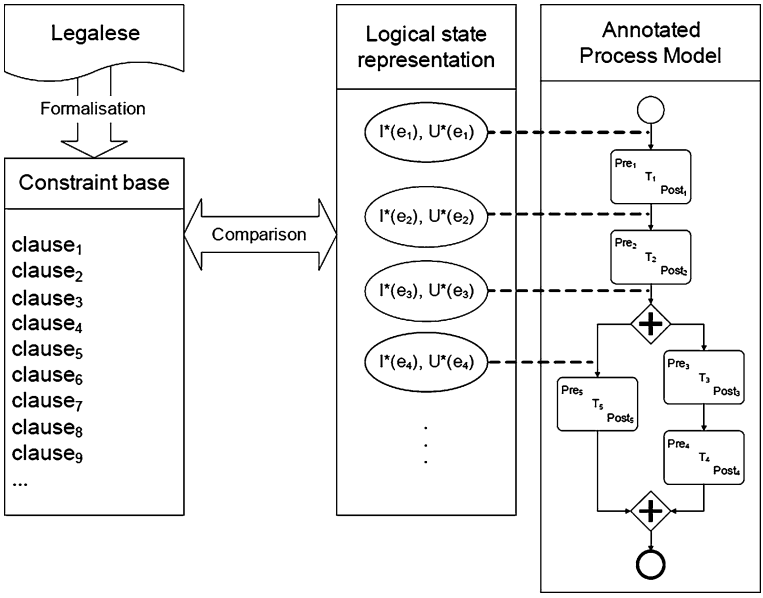


Fig. 6.6 An overview of the framework (Weber et al. 2008)

Sadiq et al. (2007) also address the problem field of business process compliance and identify the need for systematic approaches to understand the interconnection

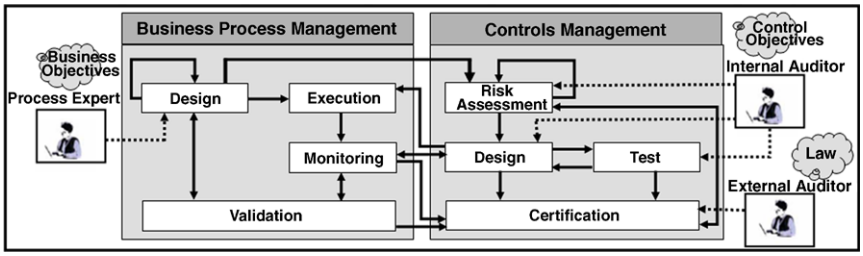


Fig. 6.7 Interconnect of process management and controls management (Sadiq et al. 2007)

and dependency between business and control objectives. Accordingly, the authors introduce a modal logic based on normative systems theory, dealing with the effective modeling of control objectives and their propagation onto business process models (Fig. 6.7).

Jallow et al. (2007) propose a framework for risk analysis in business processes with focus on cost, time and performance/quality analyses. The framework consists of the following six steps (Fig. 6.8):

1. Model the activities of the business process.
2. Determine for each activity the considered dimensions (i.e., cost, time, and output). As within a specific risk analysis only one dimension can be evaluated, the objective of each analysis has to be defined.
3. Identify risk factors, probability of occurrence, and impact.
4. Assumptions regarding the risk impact should be defined in order to consider uncertainties associated with risks. The authors use a three-point estimate expressed as triangular distribution.
5. Calculate each identified risk by multiplying the occurrence probability with the impact. "The impact is not a discrete value but a serious of values generated by the simulation based on the distribution."
6. Calculate forecasts for each activity and accumulative for the whole process.

   A prototypical framework implementation has been performed using Microsoft Excel using the add-on software Crystal Ball™.

Above, we gave a representative overview of several research approaches that aim to establish an integrated view on security, risk and business process management. It is not meant to be a holistic domain overview, but we think the selection gives the reader a good overview about developments in the recent years. Summarized, existing approaches achieved the following research results:

1. Rule-based validation of process security and selection of counter measures.
2. Extension or customizations of modeling languages (e.g., UML 2.0) by introducing security requirements modeling capabilities.
3. Stronger linkage of risk and business process management (e.g., via a taxonomy or via a reference model linking threats, vulnerabilities, ICT resources, and business processes).
4. Calculation of business process availability.

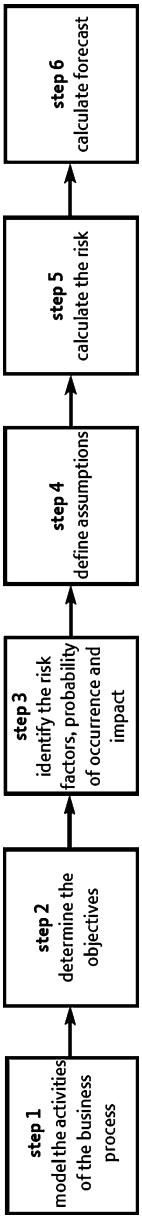


Fig. 6.8 The risk-based proposed framework (Jallow et al. 2007)

5. Integration of business and compliance objectives.
6. Determination of risk impacts on the business process activity layer using Monte Carlo simulation.

The mentioned approaches contributed substantial research in the field of business process security. However, we still miss a concept meeting the following objectives.

1. Integrated modeling concept:
   a. Business process activities
   b. Required resources
   c. Threats endangering these resources
   d. Detection, counter, and recovery measures
   e. Relations between these components
2. Concept for the simulation-based determination of risk impacts (e.g., time, costs, backlogs, etc.) on resources and/or directly business process activities considering the interaction between threats and detection-, counter-, and recovery measures.

6.3 Steps Required to Perform Risk-Aware Business Process Management

In this section we introduce the necessary steps to perform risk-aware business process management. The proposed steps must not be understood as rigid or inflexible but as requirements guidelines when setting up a respective program. The steps are derived from best-practices-guides and standards of the business process management risk management and business continuity domains. Figure 6.9 outlines the proposed steps.

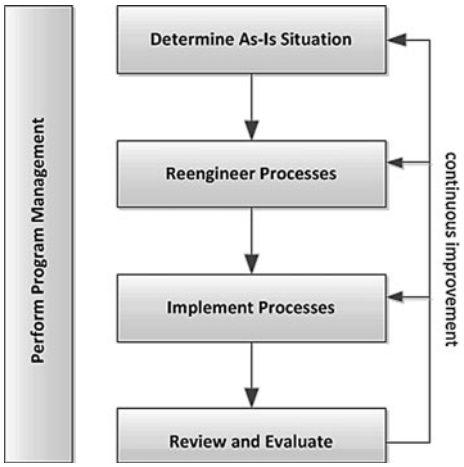


Fig. 6.9 Recommended phases for performing risk-aware business process management (Jakoubi and Tjoa 2009)

6.3.1 Perform Program Management

Within the Program Management phase the fundamentals of the planned program are established. Therefore, at least the following major topics have to be addressed:

- Scope: The Scope of the program is essential to guarantee that the program achieves the desired results. It should be clearly defined and documented which areas of an organization should be addressed by the program. Typical content of the scope definition is the identification of included business units and core processes, the geographic scale, and time and budget constraints. A good program scope definition can reduce costs. However, one should be aware that a too tight program scope definition could lead to deficiencies in the quality of results as important dependencies could be overlooked. To ensure the correctness and appropriateness of the scope, senior management should sign-off the scope of the program.
- Organizational Environment: The analysis of the Organizational Environment provides information of the vision, business objectives, and strategies of a company, as well as the market in which the company currently operates or wants to operate (e.g., competitors, customers). A clear understanding of the business forms the foundation for the evaluation of risks and the determination of mitigation strategies. The following example should clarify the statement: A company having a monopole obviously has other mitigation requirements than a company facing strong competition.
- Evaluation Criteria: In order to ensure that the program is achieving the expected goals, it is essential to introduce Evaluation Criteria. When defining these criteria, one should consider that they must be measurable in order to be evaluated. Economy-related criteria exemplarily comprise a cost reduction of ten percent, and security-related criteria a service availability of at least 99 percent.
- Roles and Responsibilities: In order to set up an effective program, Roles and Responsibilities for the program planning, execution, and controlling have to be defined. Another critical success factor for the program is senior management buy-in. It is always a good idea to have a supporting program sponsor within the board.
- Program Steering: The program coordination team is responsible for adequate Program Steering. This includes typical project management tasks such as time and budget management, quality management, and program risk management.

6.3.2 Determine As-Is Situation

The objective of this step is to gather information for further analysis steps. In order to ensure appropriate information, we recommend the following steps:

- Core Process Identification: In order to conduct risk-aware business process management, it is required to acquire the core processes of an organization. All busi-

ness units should be surveyed to ensure sufficient information about core activities, possible execution paths, and their probabilities is gathered. The business processes should be mapped to the organization's goals. Furthermore, process (activity) characteristics such as execution times and costs, and the value of the process (e.g., monetary value, intermediate products) have to be determined. Additionally, interdependencies to internal and external units should be recorded.

- Resource Identification: In this step required resources, their interdependencies, and their assignment to activities are determined. Also dependability requirements should be acquired in the resource identification phase.
- Risk Identification: The objective of this phase is to get a clear understanding about the risks a company faces. At least two types of risks should be considered when performing the risk identification: (a) Business Risks affecting process characteristics (e.g., change of invocation frequency, input parameters, change of decision probabilities). Business risks can be determined by historical data such as nonpayment of credits per year or similar key figures. (b) Resource Risks affecting dependability attributes such as confidentiality, integrity, and availability (e.g., worm disrupting the functionality of servers). The analysis of the as-is situation regarding resource related threats can be supported by tools used within the organization such as data leakage prevention solutions or event correlation tools. Furthermore, risks can be identified by using external information such as the determination of environmental vulnerability to natural disaster from meteorological institutes or information security trends from research organizations.
- Detection, Counter, and Recovery Measure Identification: The Detection, Counter, and Recovery Measure Identification deliver information about implemented measures and processes. Detection measures (e.g., fire detectors) are the basis for a successful response. Effective detection mechanisms reduce the time period until implemented counter and recovery measures may be invoked. Internal and external detection mechanisms are considered within our model. However, depending on the detection method, the initiated counter and recovery measures may vary. Counter measures can either be preventive or reactive in nature. Preventive counter measures (e.g., nonsmoking policy) reduce the occurrence probability, while reactive counter measures (e.g., fire sprinkler) decrease the potential impact by fighting the threat. Recovery measures (e.g., restore of back-up tapes) within our approach reestablish the functionality of disrupted resources.

The acquired information is modeled according to the proposed reference model, which is described later in this work, to enable further analyses such as risk-aware business process simulations as introduced in (Jakoubi et al. 2007, 2008; Goluch et al. 2008; Tjoa et al. 2008a, 2008b).

6.3.3 Reengineer Processes

The Reengineer Processes phase aims at improving the processes from an economic and from a security point of view. However, the main driver of this step is definitely

the business. Through our novel risk-aware business process simulations the risk perspective is strongly integrated in the process improvement process. The following phases have at least to be performed:

1. Business Impact Analysis: The Business Impact Analysis examines the impacts (e.g., financial, reputational) of resources' and/or activities' disruptions over time. The outputs of a business impact analysis are key figures such as the Maximum Tolerable Period of Disruption (MTPD) or the Recovery Point Objective (RPO) (Business Continuity Institute 2008).
2. Risk Analysis: The Risk Analysis identifies risks and their impact on dependability attributes of resources and/or activities. The step concludes by determining how risk should be addressed (according to the company's risk strategy) and how the process should be prioritized.
3. Identification of Improvement Options: The result of the step Identification of Improvement Options is a set of improvement alternatives for economic and security improvements. The options are presented to the senior management which has to sign-off the options that should be implemented.
4. Redesign of Processes: Once the improvement options are selected, the Redesign of Processes is performed. Secure process structures and key controls (e.g., separation of duties) should be considered while modeling the processes. The risk-aware process simulation can be used to find a proper design for the process.
5. Evaluation: The Evaluation step guarantees that the redesigned processes meet the required objectives. Deficiencies identified within this step lead to a new iteration. The new iteration can start at each process of the Reengineering Processes depending on the deficiency found. This assures the quality of the design and minimizes the threat of expensive design errors.

As described in (Jakoubi et al. 2007, 2008; Tjoa et al. 2008b), our concept of risk-aware business process modeling and simulation can be applied to support these phases.

6.3.4 Implement Processes

The Implement Processes phase aims at implementing the designed processes. Steps necessary to apply new processes to an organization comprise at least the following:

1. Project Setup: Within the Project Setup step implementation projects are set up. The roles and responsibilities for the projects are assigned, and the cost and time constraints are defined. Furthermore the clear scope of the project has to be defined, and evaluation parameter should be determined. Additionally, typical project management activities such as project controlling have to be carried out.
2. Implementation: The next step is the Implementation of the specific projects. While the implementation step it is important to evaluate the technical solutions in order to realize the design and to introduce the new processes. It is essential for the success of the project that process changes within the organization are

communicated clearly in order to improve acceptance. If necessary, awareness trainings should be carried out.
3. Evaluation: The last step of this phase is the Evaluation of the implementation. If deficiencies are identified, the issues are documented, and a new iteration can start either at the Reengineer Processes phase or at the Implementation step depending on the significance of the problem.

In general, it is hardly possible to estimate the duration of the implementation phase as it significantly depends on the approved and budgeted scope of the implementation project. However, regarding projects with duration longer than one year, it would definitely be feasible to define frequent controls of intermediate deliverables (e.g., every 3 months) to facilitate adequate project steering. Milestones with assigned deliverables should be planned at least biannually, and a greater project status evaluation should be performed at least annually.

6.3.5 Review and Evaluate

As each organization is a living entity, processes and risks have to be periodically evaluated. This ensures that processes are improved on a regular basis and that changes in risk situation are promptly recognized. Furthermore, it is essential to test and exercise the security capabilities of an organization in order to build up an efficient and effective response for unwanted events.

Applying the above described phases enables risk-aware business process management. In the following section, we present our reference model.

6.4 A Reference Model for Risk-Aware Business Process Management

In this section, we firstly introduce our reference model enabling risk-aware business process management. Later in this chapter, we outline the set of recommended business process and risk-related elements for our approach. We decided to support this specific set in order to ensure support for a broad range of modeling notations. Figure 6.10 schematically shows our reference model. Summarizing the foundation of the reference model, which can be found in (Jakoubi et al. 2007, 2008; Goluch et al. 2008; Tjoa et al. 2008a, 2008b), the concept of risk-aware business process modeling can be described as follows: Threats put business process elements (e.g., an activity or a resource) in danger. A successful attack of a threat can lead to an interruption or a delay in the execution of the business processes. In order to protect a company and its asset, three functions are required (i.e., detection, counteracting, recovery). In the following we briefly describe how we realized the functions within our reference model. Detection measures influence the time period until when counter and recovery measures are invoked. Counter measures can reduce either the

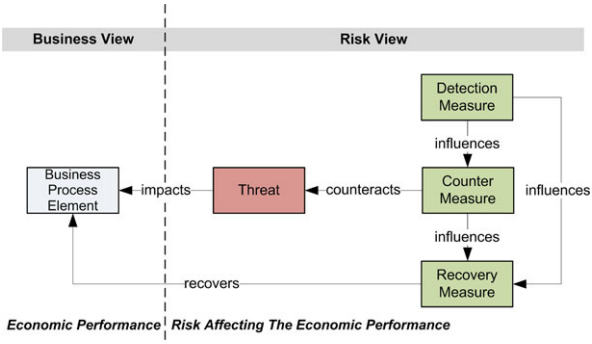


Fig. 6.10 General reference model (Jakoubi and Tjoa 2009)

likelihood of a threat's occurrence or directly counteract a threat. Recovery measures reestablish the functionality of the business process (e.g., recovery of an affected resource). Therefore our risk view contains the succeeding elements:

- Threats containing an occurrence probability affecting business process elements with a certain impact.
- Counter Measures either reduce the occurrence probability of a Threat (i.e., preventive) or the potential impact of an occurred Threat (i.e., reactive).
- A Recovery Measure reestablishes the functionality of impacted resources and/or activities.
- A Detection Measures influences the time periods until Counter and Recovery Measures are invoked.

Within our proposed approach each of the abovementioned elements can be represented as a process. In order to consider the behavior of threats, detection-, counter-, and recovery measures, we propose the usage of according functions. Exemplarily, an impact function represents the effect of a threat "fire" on the availability attribute of the resource "server room." The more expertise and historical data is at hand, the easier is the derivation of representative functions. Exemplarily, under the assumption of a severe earthquake and the absence of seismic safeguards, the determination of the threat's impact function will be straightforward. In contrast, the challenge of an "insider" threat's impact function will definitely be more difficult to solve. For assigning threats and resources, there are international standards and best-practices (e.g., BSI 2004) provide sufficient guidance. A possibility from the scientific perspective would be the usage of a security ontology (Goluch et al. 2008).

Generally our reference model does not require a specific process modeling language in order to address a broad audience. The quality of results however could vary. Therefore we recommend a minimal set of required business process elements which are outlined in Fig. 6.11. The elements on the right side (i.e., start, activity, decision, resource, resource requirements) could currently be affected by threats in our model. In order to clarify our needs we shortly describe the elements and their functionality in the succeeding paragraphs.

Within our approach, a Business Process is the container for all further elements. A Business Process can consist of the succeeding Business Process Elements:

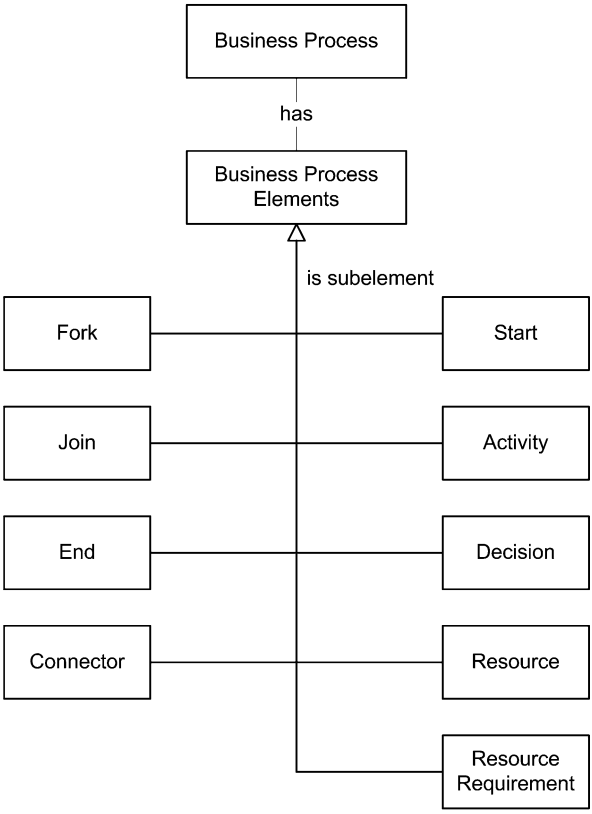


Fig. 6.11 Minimal set of business process elements (Jakoubi and Tjoa 2009)

- A Connector connects all Business Process Elements in order to describe the process flow.
- A Start is the beginning of a Business Process. There can only be one Start element. Risks that affect this element change the start parameter of the process. An example would be an increase of incoming calls within a call center. These kinds of risk will be further referred as business risk.
- An Activity transforms by definition inputs into outputs by using a specific set of resources. In order to conduct suitable analysis an activity should at least possess the economic attributes Execution Time and Costs. For risk analysis purposes, an activity should also have the following further risk-related attributes:

  1. A Completion Function which may be affected by an occurred threat. This enables us to consider delays of activities;
  2. The flag Interruptible which describes whether the execution of the activity may be delayed or the activity has to be totally reexecuted;
  3. Dependability Attributes (e.g., confidentiality, integrity, availability, etc.) stating the demand on the activity that it is correctly executed;
  4. A Priority that serves in the context of all business process activities as decision support for recovery sequences.

Risks that directly affect an activity threaten the continuous or correct execution of an activity. Examples would be accidental human erratic behavior caused by lack of knowledge.

- A Resource is required to perform activities. A Resource has at least the economic attribute Cost. Furthermore, it has a Type (e.g., input or output) and Dependability Requirements stating the demand on the resource that it can be correctly used. Risks can affect the attributes of resources such as the availability of resources. Examples of threats that could affect resources are an aggressive worm or an earthquake which may disrupt the functionality of resources. Risks affecting the resources will be further referred as term Resource Risk.
- A Resource Requirement describes the interrelationship between an Activity and a set of Resources. The attribute Dependability Level states the demand of an Activity which has to be met by the resource (e.g., Resource A must be fully available). The attribute Logical Connection relates resources (e.g., logical operators AND or OR) in order to exemplarily represent redundancies. Typical risks affecting this element are business risks such as peak periods or incorrectly planned resource needs may affect this element's characteristic.
- A Decision splits the process flow into at least two branches. The attribute Threshold describes how branches are chosen. Typically, each branch has a certain probability that it will be chosen during a simulation. However, other constraints such as monetary values (e.g., lower than or greater than amount X) are possible. Business risks may affect the probability distribution of outgoing edges.
- A Fork splits the process flow into at least two branches which are parallel executed.
- A Join is assigned to a specific Fork in order to unite the parallel executed process paths.
- An End marks that the process execution stops at this point. More than one End is possible.

The comprehensive information of business process elements under consideration of all risk-related elements enables the determination (e.g., via simulation) of the processes' performance.

All Business Process subelements can be integrated in the left-sided Business View (Fig. 6.11). However as mentioned above, only the Business Process subelements Start, Activity, Decision, Resource, and Resource Requirement can be attacked by threats (right-sided Risk View). Figure 6.12 shows as demonstrative example the integration of the subelements Activity, Resource, and Resource Requirement and the interconnection between the Business and Risk View.

6.5 Application Scenarios

In this section we first want to outline two business cases in order to show the capabilities of our approach. Secondly, we outline further application scenarios of our approach with a special focus on resource utilization. For the sake of clarity,

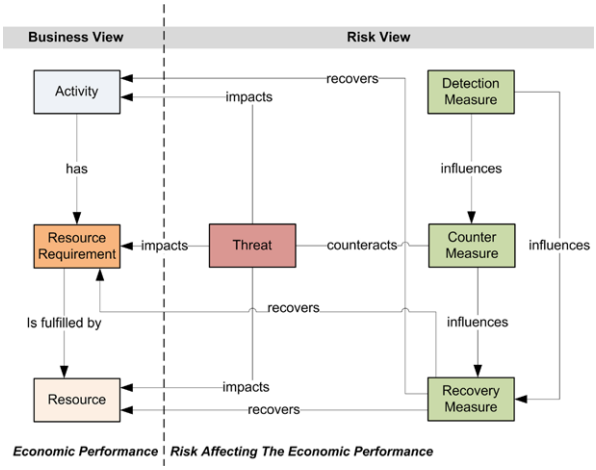


Fig. 6.12 Reference model applied on the business process elements activity, resource requirement and resource (Jakoubi and Tjoa 2009)

these are stylized use cases. Figure 6.13 gives a conceptual overview about our approach which serves as basis for two demonstrative service level analysis scenarios within the company ACME: (1) A threat (T1) endangers the assigned resource and (2) the outage of the underpinning contract (UC) shall be evaluated in the course of a what-if simulation. In scenario 1, a threat puts an assigned resource (R) in danger. In order to better demonstrate the effects of a threat our resource model indicates that R is not redundantly sized but required (i.e., logical and relation). A disruption of the resource would therefore cause a delay in the execution of the business process activity 2 (Act 2). In a nutshell, detection measures try to detect an occurred threat and invoke corresponding counter measures which try to eliminate the threat. Subsequently—or partially overlapping—recovery measured try to restore the disrupted resource. The bottom line is that in the course of the simulation, the threat is eliminated and the resource restored. Thus, Act 2 can be again executed. Through our risk-aware business process simulation, we are able to determine additional costs and times through invoking detection-, counter-, and recovery measures and to consider delays or total outages in the activity's execution. This can consequently be used to analyze signed Service Level Agreements. One example outcome is the probability that—on the basis of the modeled company's as-is situation and one or more threat scenarios—the signed agreement will be breached leading to arising penalties for ACME. In scenario 2, an underpinning contract (UC) is analyzed. There, three interesting questions for ACME are: (1) "to which availability extent the agreed service is required?", (2) "what are the impacts of a UC outage (i.e., contract breach of ACME's service provider)?", and (3) consequently, "which penalty has to be agreed to adequately transfer the financial risk?". In the modeled as-is situation, there is no continuity plan for UC implemented, thus it is according to the resource model completely required to perform Act 2. Applying our risk-aware business process simulation, ACME can simulate various contract options (e.g., bronze level with 90 percent guaranteed availability, silver level with 99 percent availability, and gold level with 99.99 percent availability). Consequently, the

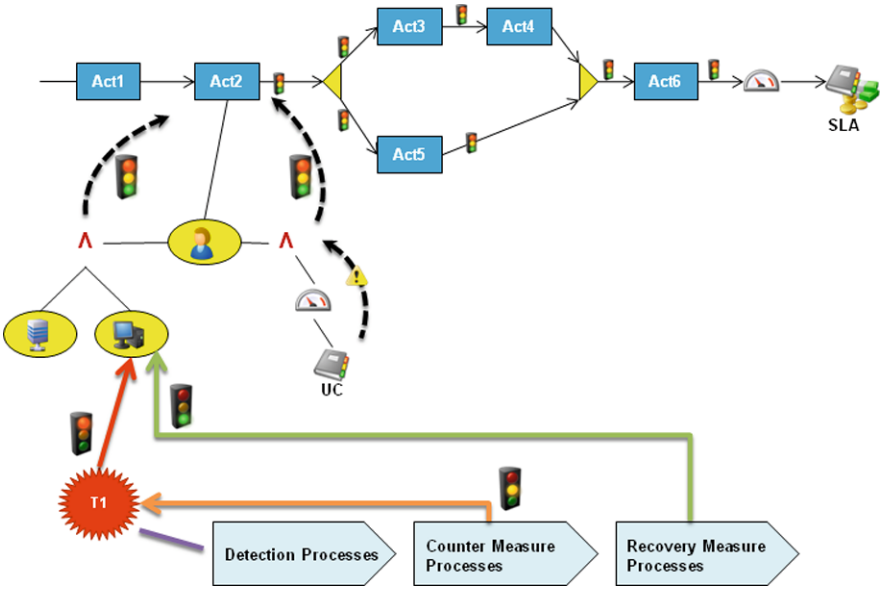


Fig. 6.13 Conceptual risk-aware service level analysis

simulated situations can be evaluated for one business year taking at least contract costs, outage probabilities, possible business process executions paths (e.g., through decision elements), peak periods (e.g., at the end of the accounting year), and resulting (financial) impacts into account.

We implemented a prototype within the Matlab® module Simulink® (The MathWorks 2010). The following figures sketch extracts from our simulation results for scenario 2. The bronze level selection would cause significant backlogs (see Fig. 6.14) and would lead to potential service level breaches for ACME under the assumption that the vertical dotted line is the evaluation baseline. In comparison to the selection of the silver level (see Fig. 6.15), the results support decision makers to choose the silver level. Further investments to buy the more comprehensive gold level are questionable as the silver level seems to be sufficient.

In the following we want to outline further application scenarios which focus on resource utilization and are described in detail in Jakoubi et al. (2008).

- Simulation-based determination of resources working capacities (in percent) in case of reallocation between processes, for example, a resource is required for 100 percent by its dependent business process and simultaneously for 40 percent by a threat impact process. Thus, its theoretically required working capacity is 140 percent.
- Simulation-based determination of the changing resource utilization during threat scenarios, for example, the reallocation of personnel from a business process in order to counteract an occurred threat affecting the operability of another (e.g., higher prioritized) business process.

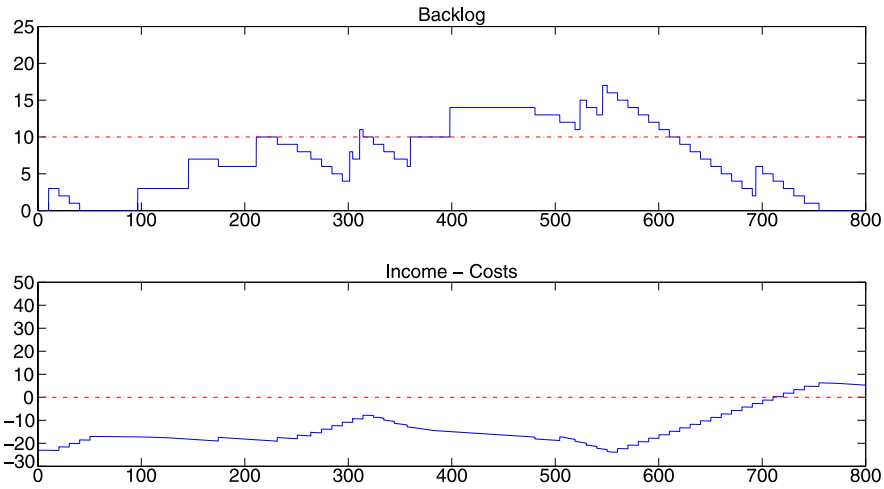


Fig. 6.14 Simulation result excerpt: bronze level

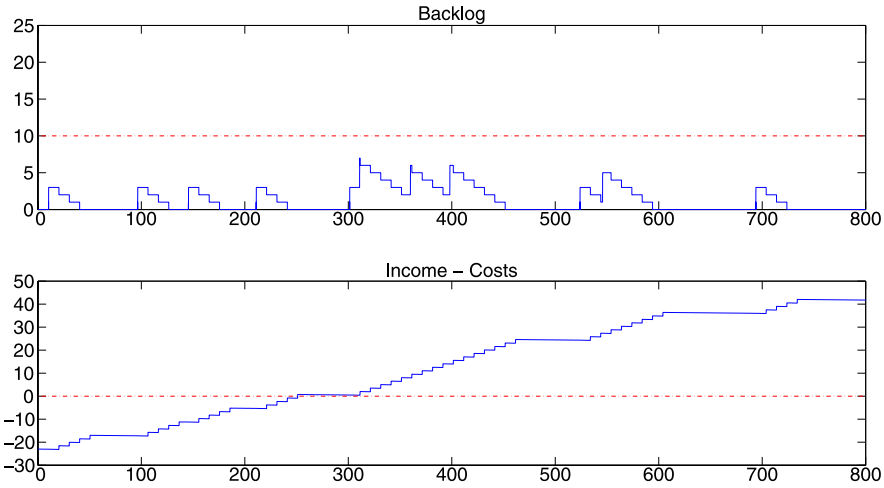


Fig. 6.15 Simulation result excerpt: silver level

- Simulation-based determination of additional costs, which are caused by this changing resource utilization. For instance, personnel have to be reallocated to guarantee the operability of a higher prioritized business process. As a consequence, the execution of the lower prioritized business process is disrupted leading to resources not working to full capacity.
- Simulation-based determination of resource requirements to minimize the impact of an occurred threat considering the shortage of resources resulting from downtimes of resources or insufficient resource capacities.

- Simulation-based identification of essential resources, which would cause severe backlogs of their dependent processes in case of reallocations.
- Simulation-based determination of additional resource requirements to eliminate backlogs caused by occurred threats.

All in all, in this section we outlined how our recently introduced concept of risk-aware business process simulation can be used to analyze security and economic viewpoints of business process. We believe that our approach brings significant benefits by using synergy effects of the business process, business risk, and business continuity domains.

6.6 Conclusion

The execution of business processes is the fundament of a company to meet its business objectives. These business processes are either support processes or directly provide aimed results (e.g., a product or a service for a customer). Business process management is the dominant domain aiming at optimizing the execution of business processes so that activities are performed efficiently and effectively in economic terms. "The biggest benefit of business process optimization and simulation is that they deliver insight into dynamic processes so that they are designed well and operated effectively as conditions change" (Gartner Inc. 2009). However, business processes face threats that endanger the effective and efficient execution of their activities. There exist diverse classifications of these threats (National Institute of Standards and Technology 2002; BSI 2004; International Organization for Standardization 2004) ranging from accidents (e.g., unavailability of ICT resources or the absence of strategic personnel) to natural catastrophes (e.g., earthquakes), and to deliberate acts (e.g., sabotage or theft). The reasons why the execution of business processes may be affected causing negative effects on business are manifold and addressed by several domains. Traditional risk management—and thus implementations in according tools—considers risks on business process in a rather static way. Dynamic aspects are in fact only included through organizational risk management processes that let assessment be reperformed in certain intervals (e.g., biannually or annually). With our approach, we try to overcome this shortcoming and to use the advantage of dynamic business processes analysis when incorporating risk aspects leading to significant domain-overleaping synergy effects. Within this chapter we presented a reference model enabling the consideration of risks within business process evaluations based on our previously conducted research. Through the description of the requirements needed to enable risk-aware business process simulation, we are independent of graphical notation. As long as mandatory components and relations are considered, an evaluation is possible. Main benefits arising from the application of our approach comprise:

- Integrated modeling of business processes, risks, and detection, counter, and recovery measure information. Consequently, this allows the simulation of threats

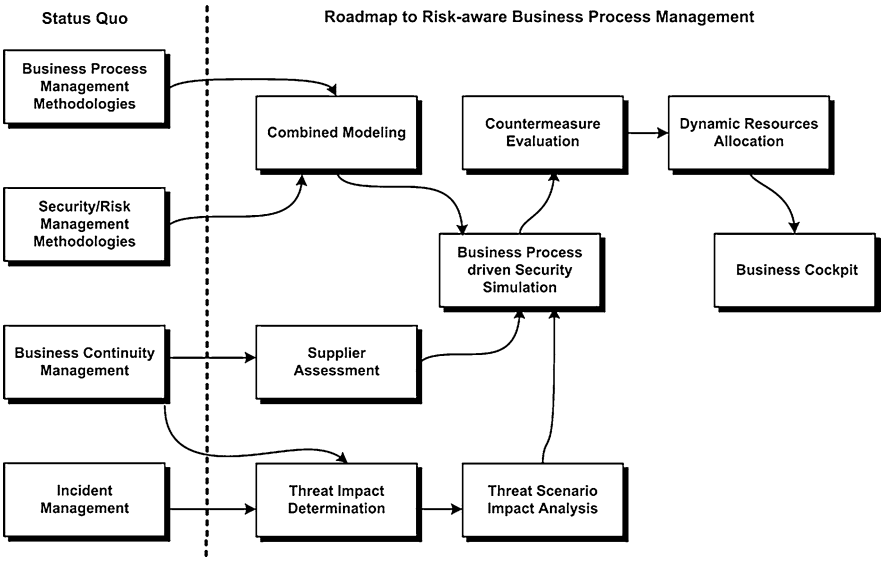


Fig. 6.16 Roadmap to risk-aware business process management (Jakoubi et al. 2009)

and safeguard measures on attributes of business process elements, such as the availability or integrity of a resource. Subsequently, impacts on business process executions can be derived in a simulation-based way.

- Modeling and simulation of manifold scenarios to enable an evaluation of different security/contingency solutions.
- Identification of single points of failure or substantial weaknesses in resource planning and allocation. Simulation-based determination of resource requirements of business processes with regard to numerous threat scenarios.
- Provision of valuable information concerning the justification of security/contingency investments when simulating different threatening and mitigation scenarios. Metrics such as the maximum tolerable period of disruption (MTPD) or mean time between failures (MTBF) can easily be determined. These may again serve as valuable input, e.g., for reviewing service level agreements.
- Simulation-based support of target-performance evaluations enhancing continuous process improvement cycles.
- Resource utilization strategies considering risks.

The presented formal model is a first step enabling the simulation-based evaluation of business process security. Figure 6.16 gives an overview about necessary steps towards comprehensive risk-aware business process management.

Giving a future outlook, the authors' next research efforts will be laid in the inclusion of a dynamic reallocation of resources (e.g., for reducing backlogs caused by a threat's impact) and in the in-depth consideration of service level management aspects leading to risk-aware service level planning and analysis.

References

F. Braber, I. Hogganvik, M.S. Lund, K. Stølen, and F. Vraalsen. Model-based security analysis in seven steps—a guided tour to the CORAS method. *BT Technology Journal*, 25:101–117, 2007.

British Standard Institute (BSI). British standard bs25999-1:2006: Business continuity management—part 1: Code of practice, 2006.

British Standard Institute (BSI). British standard bs25999-2:2007: Business continuity management—part 2: Specification, 2007.

BSI (German Federal Office for Information Security). IT-Grundschutz Manual (English version), 2004.

Business Continuity Institute. Good Practice Guidelines, 2008.

A. Ekelhart, S. Fenz, and T. Neubauer. Aurum: A framework for supporting information security risk management. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICCS 2009)*, pages 1–10, 2009a.

A. Ekelhart, S. Fenz, and T. Neubauer. Ontology-based decision support for information security risk management. In *International Conference on Systems (ICONS 2009)*, pages 80–85, 2009b.

European Commission. Auditing directives. URL: http://ec.europa.eu/internal_market/auditing/directives/index_en.htm, Accessed May 2010.

European Network and Information Security Agency (ENISA). Business and it continuity overview and implementation principles, 2008.

S. Fenz, A. Ekelhart, and T. Neubauer. Business process-based resource importance determination. In *Proceedings of the 7th International Conference on Business Process Management (BPM2009)*, pages 113–127, 2009.

Gartner Inc. Gartner EXP worldwide survey of more than 1500 CIOS shows IT Spending to be flat in 2009, 2009.

G. Goluch, A. Ekelhart, S. Fenz, S. Jakoubi, S. Tjoa, and T. Mück. Integration of an ontological information security concept in risk aware business process management. In *41st Hawaii International Conference on Systems Science (HICSS-41 2008)*, page 377, 2008.

Gartner Inc. Misconceptions on process optimization and simulation. Gartner Blog, 2009.

International Organization for Standardization. Iso/iec 13335-1:2004, information technology—security techniques—management of information and communications technology security, Part 1: Concepts and models for information and communications technology security management, 2004.

International Organization for Standardization. Iso/iec 24762:2008 information technology—security techniques—guidelines for information and communications technology disaster recovery services, 2008.

S. Jakoubi and S. Tjoa. A reference model for risk-aware business process management. In *International Conference on Risks and Security of Internet and Systems*. IEEE, New York, 2009.

S. Jakoubi, S. Tjoa, and G. Quirchmayr. Rope: A methodology for enabling the risk-aware modelling and simulation of business processes. In *Fifteenth European Conference on Information Systems*, pages 1596–1607, 2007.

S. Jakoubi, G. Goluch, S. Tjoa, and G. Quirchmayr. Deriving resource requirements applying risk-aware business process modeling and simulation. In *16th European Conference on Information Systems*, pages 1542–1554, 2008.

S. Jakoubi, T. Neubauer, and S. Tjoa. A roadmap to risk-aware business process management. In *Proceedings of the International Workshop on Secure Service Computing (SSC 2009)*, 2009.

A.K. Jallow, B. Majeed, K. Vergidis, A. Tiwari, and R.Roy. Operational risk analysis in business processes. *BT Technology Journal*, 25:168–177, 2007.

D. Karagiannis, J. Mylopoulos, and M. Schwab. Business process-based regulation compliance: The case of the sarbanes-oxley act. In *Proceedings of the 15th IEEE International Requirements Engineering Conference*, pages 315–321, 2007.

N. Milanovic, B. Milic, and M. Malek. Modeling business process availability. In *IEEE International Conference on Services Computing (SCC 2008)*, pages 315–321, 2008.

National Institute of Standards and Technology. NIST SP800-30, risk management guide fir information technology systems, 2002.

National Institute of Standards and Technology. NIST SP800-61: Computer security incident handling guide, 2004.

D. Neiger, L. Churilov, M. zur Muehlen, and M. Rosemann. Integrating risks in business process models with value focused process engineering. In *European Conference on Information Systems (ECIS 2006)*, 2006.

One Hundred Seventh Congress of the United States of America. Sarbanes–Oxley Act, 2002.

A. Rodríguez, E. Fernández-Medina, and M. Piattini. Towards a UML 2.0 extension for the modeling of security requirements in business processes. In *International Conference on Trust and Privacy in Digital Business (TrustBus 2006)*, pages 51–61, 2006.

S. Sackmann. A reference model for process-oriented IT risk management. In *16th European Conference on Information Systems*, 2008.

S. Sackmann, L. Lowis, and K. Kittel. Selecting services in business process execution—a risk-based approach. In *Business Services: Konzepte, Technologien, Anwendungen, Tagung Wirtschaftsinformatik (WI09)*, 2009.

S. Sadiq, G. Governatori, and K. Namiri. Modelling control objectives for business process compliance. In *5th International Conference on Business Process Management (BPM2007)*, pages 149–164, 2007.

The MathWorks. Simulink—simulation and model-based design, URL: http://www.mathworks.com/products/simulink/, Accessed May 2010.

S. Tjoa, S. Jakoubi, G. Goluch, and G. Quirchmayr. Extension of a methodology for risk-aware business process modeling and simulation enabling process-oriented incident handling support. In *Advanced Information Networking and Applications*, pages 48–55, 2008a.

S. Tjoa, S. Jakoubi, and G. Quirchmayr. Enhancing business impact analysis and risk assessment applying a risk-aware business process modeling and simulation methodology. In *International Conference on Availability, Reliability and Security*, pages 179–186, 2008b.

I. Weber, G. Governatori, and J. Hoffmann. Approximate compliance checking for annotated process models. In *1st International Workshop on Governance, Risk and Compliance—Applications in Information Systems (GRCIS'08)*, 2008.

M. zur Muehlen and M. Rosemann. Integrating risks in business process models. In *Australasian Conference on Information Systems (ACIS 2005)*, 2005.

Chapter 7
Self-Optimised Tree Overlays Using Proximity-Driven Self-Organised Agents

Evangelos Pournaras, Martijn Warnier, and Frances M.T. Brazier

Summary Hierarchical structures are often deployed in large-scale distributed systems to structure communication. Building and maintaining such structures in dynamic environments is challenging. Self-organisation is the approach taken in this chapter. AETOS, the Adaptive Epidemic Tree Overlay Service, provides tree overlays on demand. AETOS uses three local agents to this purpose (i) to translate application requirements to self-organisation requirements, (ii) to self-organise nodes into optimised tree topologies based on these requirements, and (iii) to control bootstrapping and termination of self-organisation. The evaluation of AETOS in different simulation settings shows that it provides high connectivity in tree overlays optimised according to application requirements.

7.1 Introduction

Complex, intelligent, distributed systems in dynamic environments need to adapt continuously. Management is a challenge. Central management of such systems is not often an option: distributed management is required.

Self-management relies on local management at the level of individual systems, and virtual topologies (overlays) to regulate communication between systems, for example to aggregate global knowledge about the state of a system. Hierarchies often provide the structure upon which distributed management is based. Examples of

E. Pournaras (✉) · M. Warnier · F.M.T. Brazier
Department of Multi-actor Systems, Section Systems Engineering, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands
e-mail: e.pournaras@tudelft.nl

M. Warnier
e-mail: m.e.warnier@tudelft.nl

F.M.T. Brazier
e-mail: f.m.brazier@tudelft.nl

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,* Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_7, © Springer Science+Business Media, LLC 2010

domains of applications for which this holds include DNS, multimedia multicasting (Tan et al. 2006), energy management (Pournaras et al. 2009a) and distributed databases (González-Beltrán et al. 2008).

Building and maintaining robust and application-independent hierarchical topologies designed to this purpose is the challenge this chapter addresses, in particular for tree structures. Connectivity in a tree overlay is of key importance. If a node is (temporarily) disconnected, the branches underneath the node are also (temporarily) disconnected from the rest of the system, affecting global performance.

AETOS, the Adaptive Epidemic Tree Overlay Service, is the approach proposed in this chapter. AETOS makes it possible to create self-organised tree topologies that are proactively resilient to failures, and reactively self-heal (Chaudhry and Park 2007) the structure built. AETOS (Pournaras et al. 2009b) builds and maintains application-independent robust tree topologies in dynamic distributed environments.

Intelligent software agents are used (i) to translate application requirements to self-organisation requirements, (ii) to self-organise nodes in optimised tree topologies based on these requirements, i.e., reactively reconnecting or rewiring connections to improve robustness, and (iii) to control bootstrapping and termination of self-organisation.

Experimental evaluation of the AETOS self-organisation based on connectivity convergence is presented.

This book chapter is outlined as follows: Sect. 7.2 outlines application domains in which hierarchical topologies are used. It also illustrates the problem and summarises the contributions of AETOS. Section 7.3 illustrates related work on robust tree overlays. Section 7.4 provides a high-level overview of the agent-based approach of AETOS. Sections 7.5–7.7 present the three agents of AETOS: the 'application agent', the 'self-organisation agent' and the 'system control agent' respectively. Section 7.8 illustrates the experimental evaluation of the approach that this book chapter proposes. Finally, Sect. 7.9 concludes this chapter and outlines future work.

7.2 Objectives and Contributions

This section discusses the importance of tree topologies for various application domains and identifies the problem of managing application-independent self-organised trees. It also provides an overview of the proposed solution.

7.2.1 Applications

Tree structures are often used in information management for aggregation, search, dissemination and decision-making. Their complexity is usually bounded to a logarithmic function or to the number of nodes in the tree structure. They are also used

for many other purposes, such as knowledge extraction and visual information systems.

Although the use of trees in centralised systems is typical and has been extensively studied, using and maintaining a tree structure in a decentralised system is the challenge this chapter addresses. Introducing a dynamic tree structure for distributed systems potentially enables effective self-management. As an example, EPOS, the Energy Plan Overlay Self-stabilisation system (Pournaras et al. 2009a), performs stabilisation in the global energy utilisation of thermostatically controlled devices. These devices are interconnected and organised in a tree overlay. Based on this structure, they perform local aggregation and decision-making of the local allocated energy they consume for a period of time. EPOS achieves the minimisation or the reverse of the deviations in the global energy utilisation making it possible (in theory) for power systems to become more robust and flexible to dynamic environments.

IP multicast appears to have many limitations in its adoption and deployment (Diot et al. 2000), especially concerning the average end user. These limitations are related to its routing complexity and scalability. Application-level multicast has emerged as a new approach for distributing multimedia content. The majority of methods based on application-level multicast use tree overlays. Organising nodes in a loop-free structure can make distribution of content effective and potentially scalable compared to mesh-based overlays. Extensive comparisons of various application-level multicast approaches are illustrated in Birrer and Bustamante (2007), Liu et al. (2008), Tan et al. (2006).

Tree structures integrated with skip lists (Pugh 1990) in skip tree graphs benefit distributed database operations such as range queries (González-Beltrán et al. 2008). In the same domain, tree overlays, introduced as a distributed indexing scheme, enhance resource searching and sharing (Zhuge and Feng 2008). Finally, super-peer topologies model distributed systems in a hierarchical fashion that can reflect the heterogeneity of different node capabilities, such as storing capacity, processing power, connectivity or bandwidth. This provides the potential for various application optimisations, such as load-balancing. Such an option is explored in ERGO, the Enhanced Reconfigurable Gnutella Overlay (Pournaras et al. 2008). ERGO rewires nodes with high outgoing load to nodes with low incoming load. This is achieved through the interaction of lower-level nodes with higher-level *virtual server* nodes responsible for load-balancing.

7.2.2 Problem Statement

As stated above, building and especially maintaining tree overlays, optimised for different applications, is the problem this chapter addresses. The main aspects of this problem are: self-organisation, self-optimisation, and application independence.

Self-Organisation Nodes should be able to self-organise themselves in a tree overlay using local knowledge. Often, as explained in Sect. 7.6, this knowledge

is a partial view of the distributed environment. Nodes should be able to connect to other nodes and potentially rewire connections without introducing loops or violating restrictions such as their capacity.

Self-Optimisation The satisfaction of application requirements when using tree overlays is usually an optimisation problem, as described in detail in Sect. 7.3. Nodes should connect to the appropriate neighbours to maximise their applications' utilities. Note that applications most often use topologically different tree overlays as they are based on different performance metrics. For example, in EPOS (Pournaras et al. 2009a) availability of nodes is the metric used to identify disconnected nodes. Note that availability is a metric measurable in many applications, such as Overnet (Bhagwan et al. 2003). Similarly, application-level multicast tree overlays are based on metrics such as latency, bandwidth, node degrees and other. Section 7.3 discusses related approaches.

Application Independence Providing a dedicated self-organisation mechanism for each application can be costly. Distributed systems are dynamic and support applications that interact with each other.

Figure 7.1 illustrates the concept of different tree overlays on the same physical network. Each overlay is used by a different application. A physical host corresponds to one (or more) overlay host in an overlay network. Note that the position of an overlay host in a tree overlay is different for each application overlay. This is because the mapping between a physical host and the respective overlay hosts depends on the application requirements and optimisation metrics.

Each application, for every overlay, is responsible for building and maintaining the tree structure. A generic self-organisation middleware service can decouple the building and maintenance from the application. This chapter focuses on the problem of how such a service can be modelled and how it can function in large-scale distributed systems, such as virtual networks over physical infrastructures or large-scale multiagent systems.

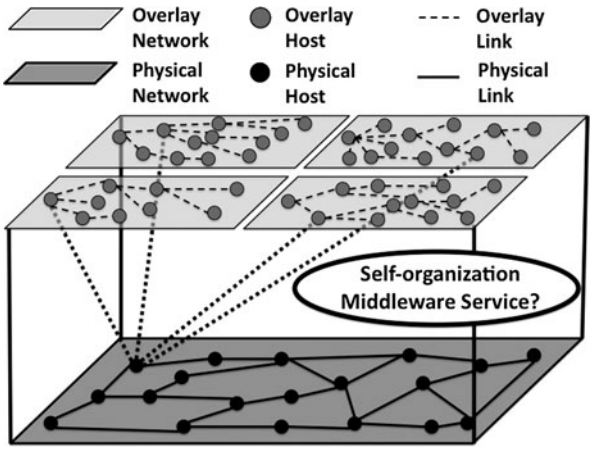


Fig. 7.1 Each overlay application requires a different optimised tree topology. A physical node corresponds to overlay nodes placed in different position in the tree. In this case, the building and maintenance is application-dependent. A self-organisation middleware service for tree overlays could solve this problem

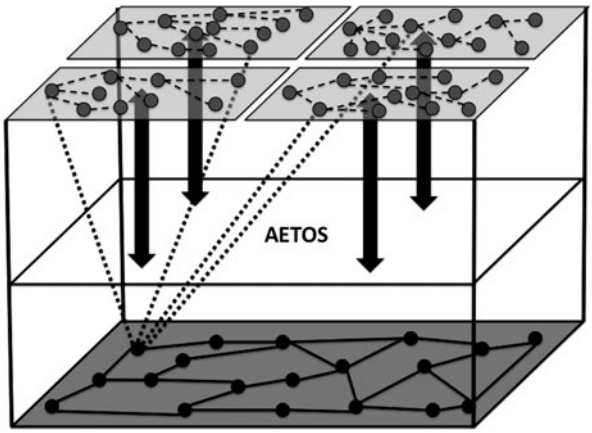


Fig. 7.2 AETOS is placed as a middleware service in a distributed environment. It undertakes the role of building and maintaining different tree overlays for different applications

7.2.3 System Overview

The contribution of this chapter is to propose a self-organisation service for tree overlays, named AETOS, the Adaptive Epidemic Tree Overlay Service. AETOS is an agent-based system positioned between the overlay applications and the physical network. Figure 7.2 illustrates the position and the interactions of AETOS in a distributed environment.

Overlay nodes have direct access to information from the physical network. This information, together with other application requirements, is passed to the AETOS layer. Based on this information, AETOS builds and maintains on-demand different tree overlays for each application.

AETOS achieves the abstraction of local application requirements to local self-organisation requirements. Nodes are dynamically self-organised to tree topologies on-demand based on their proximity derived from the local application requirements. Bootstrapping and termination of self-organisation is managed locally.

The experimental evaluation of Sect. 7.8 reveals that AETOS achieves high connectivity of tree overlays in various experimental settings. This chapter also investigates the influence of various factors in the cost-effectiveness of AETOS.

7.3 Related Work

This section presents related literature on self-organised and robust tree overlays, focusing in particular on: (i) *application domain*, (ii) *optimisation metrics*, (iii) *complementary overlays*, (iv) *build and maintenance*, (v) *decentralisation level*, and (vi) *proactiveness vs. reactiveness*. Open issues are discussed and outlined, illustrating the need for a self-organisation service, such as AETOS.

7.3.1 Literature Review

This section provides an overview of related work in the area of robust self-organised tree overlays, on the basis of the six areas distinguished above.

Application Domain The majority of the methods concerns tree overlays for application-level multicast, video streaming, and real-time applications, as underlined in Sect. 7.2.1. Multimedia applications require effective broadcast for guaranteeing high QoS. The root is usually the provider of the multimedia content, and the rest of the nodes are end-users that receive this content. They contribute resources in the system by forwarding the content they receive from their parents to their children. In database systems, complex queries can be performed over peer-to-peer tree overlays (Jagadish et al. 2006; Li et al. 2006). Maintaining a robust and reliable topology is crucial for data consistency and knowledge extraction from the network. Publish–subscribe systems (Costa and Frey 2005; Frey and Murphy 2008) also benefit from tree overlays as they can be used to minimise the changes in the event routing. Other domains in which tree overlays are deployed are grid environments for task allocation and scheduling (Chakravarti et al. 2005; England et al. 2007) and sensor networks for data collection (England et al. 2007). Note that, although these applications vary significantly and have different requirements, the common goal of all of them is shared: to maximise its utility by performing operations over application specific optimised tree overlays.

Optimisation Metrics Robustness in tree overlays can be achieved by single- or multi-metric objective optimisations in the self-organisation process. Various optimisation metrics, related to the application type, are used to organise nodes in an appropriate tree overlay for the application. Some of the most common optimisation metrics include delay, bandwidth, node degree, uptime, and other related optimisation metrics. Note that these metrics are usually related to the underlying physical network, in order for applications to maximise the utilisation of available network resources. In England et al. (2007), trees are optimised by considering the number of hops and the eccentricity, both metrics related to the experienced delay in the underlying physical network. Bandwidth and node degree are associated in Fei and Yang (2007). The number of children influences the bandwidth consumed from their parents in multicasting applications. In contrast, these two metrics are assumed independent in mTreebone (Wang et al. 2007). This assumption is valid when other applications consume part of the available bandwidth. Node degree influences the topology and the optimisation of the application. Trees can be balanced, fat (wide), or long ones. Tree topologies, such as the latter two ones, can be integrated by exploiting trade-offs between opposing performance metrics, i.e., uptime and bandwidth (England et al. 2007; Tan et al. 2006). Similar trade-offs are explored in Li and Ooi (2004) as well. In these cases, multi-metric objective optimisations are applied by combining or weighting two or more metrics. For example, in Tan et al. (2006), bandwidth and

uptime are combined by computing their product, the 'service capability contribution'. Weighting schemes between 'path weight'–'hop count' and 'delay penalty'–'resource usage' are proposed in England et al. (2007) and Li and Ooi (2004) respectively. Finally, the sojourn probability (Lee and Kim 2007) and the joining times of nodes (Liu and Zhou 2006) can be used for the optimisation of tree overlays.

Complementary Overlays Some multicast applications maintain tree overlays over mesh ones. RESMO (Li and Ooi 2004) is a minimum delay, minimum resource usage spanning tree over a mesh overlay. RESMO selects links from the mesh overlay with sufficient bandwidth. mTreeBone (Wang et al. 2007) is based on the similar concept of selecting stable nodes from the mesh overlay to build a backbone tree overlay. MeshTree (Tan et al. 2005) is a combination of a tree and mesh overlay by inserting shortcut links between the nodes of the tree overlay. Such a link redundancy is used in other approaches as well. For example, BATON* (Jagadish et al. 2006) additionally inserts adjacent and neighbour links between nodes of the tree for acquiring additional robustness. TAG (Liu and Zhou 2006) and PRM (Banerjee et al. 2006) use gossiping and random links respectively to deal with data loss and discontinuous playback in real-time applications. Gossiping is used to support trees in GoCast (Tang and Ward 2005) as well. Other underlying complementary overlays that appear in literature are DHTs (Costa and Frey 2005). However, DHTs are not resilient to failures and require maintenance.

Build and Maintenance In the investigated approaches in this section, the building process of tree overlays is either integrated with their maintenance, for example in Leitao et al. (2007), or it serves as a bootstrapping mechanism for the maintenance that follows, e.g. (Banerjee et al. 2003). The main method used for building a tree, or an initial version of it, is the consecutive joins to candidate parents and children (Tan et al. 2005) or to the leaves of the tree (Tan et al. 2006). These candidates are derived randomly (Lee and Kim 2007) or from their proximity to the local node (Liu and Zhou 2006). After the initial joins, nodes either aim to improve their position in the tree or cooperate to optimise the tree topology. In the first case, nodes perform shift-up operations (Tan et al. 2006) by moving to an upper level in the tree, whereas, in the second case, a parent and one of its children swap their positions (Akbari et al. 2005; Jagadish et al. 2006). Plumtree (Leitao et al. 2007) combines eager and lazy push gossiping strategies to build and maintain a tree overlay. In Costa and Frey (2005), the node-key mapping of the underlying DHT is used to form the tree overlay. Alternative methods for the distributed building of a tree overlay include the top-down approach proposed in Li and Ooi (2004), the Bellman Ford (England et al. 2007) and Prim's algorithm (Fei and Yang 2007). Furthermore, nodes can monitor the connectivity of their neighbours by sending heartbeats (Li et al. 2006; Liu and Zhou 2006). In case of a failure, they try to connect with another node. TreeOpt (Merz and Wolf 2007) improves the tree connectivity by performing two types of children moves as an evolutionary optimisation of the tree overlay. In Frey and Murphy (2008), a candidate parent is selected by applying and combining different repair strategies related to the application requirements. Similarly in Chakravarti et al. (2005), ancestor lists are retained in case of failures.

In contrast, the proposed approach in Fei and Yang (2007) defines a 'parent-to-be' for every node (besides the root) before a failure occurs. Thus, repair is faster. Other techniques propose link redundancy in order to satisfy alternative connectivity in case of failures (Jagadish et al. 2006; Wang et al. 2007). Load-balancing also supports the maintenance of tree overlays by aiming to retain the load in the nodes between root and leaves equal (Jagadish et al. 2006; Li et al. 2006).

Decentralisation Level Among the illustrated approaches, there are some hybrid schemes for topology management. DPOCS (Akbari et al. 2005) is based on the 'overlay control server (OCS)' that assists nodes to join the multicast groups. OMNI (Banerjee et al. 2003) and TAG (Liu and Zhou 2006) follow a similar concept by introducing the 'multicast server nodes (MSNs)' and a 'content server' respectively. mTreebone (Wang et al. 2007) utilises only stable nodes for video multicasting. BulkTree (An et al. 2006) groups the nodes to 'super-nodes' in order to increase the stability of the tree. Finally, the approach of Lee and Kim (2007) is based on a video broadcasting source node that centrally collects and calculates statistics. This information is used during for the self-organisation process.

Proactiveness vs. Reactiveness Methods that apply a sorting of the nodes, within the tree overlay, for application optimisation are considered proactive. For example, the use of 'service capability contribution' (Tan et al. 2006) as a metric for combining a bandwidth-ordered and a time-ordered tree makes the multicasting proactively more robust and efficient. Methods that use complementary overlays (Leitao et al. 2007; Tan et al. 2005; Tang and Ward 2005; Wang et al. 2007), link and data redundancy (Banerjee et al. 2006; Jagadish et al. 2006; Li et al. 2006; Liu and Zhou 2006) are also regarded as proactive approaches. In this case, proactiveness is applied indirectly and externally, by other overlay support. In Fei and Yang (2007), a highly proactive approach is proposed. Nodes calculate the new parents for their children before a failure occurs and without violating the node degrees. In contrast, reactive nodes monitor their neighbours (Li et al. 2006) and perform reconnections to other nodes when a failure occurs. Usually the selection of the nodes is based on various strategies (Frey and Murphy 2008) that balance performance trade-offs. TAG (Liu and Zhou 2006) can be considered to be a reactive system as it operates in highly dynamic environments with real-time constraints. Proactive approaches benefit from the fact that they aim to decrease the complexity and time of the repair actions or the impact of failures. However, proactive approaches introduce: (i) a usually constant but (ii) significant communication and processing cost.

7.3.2 Open Issues

The conclusions from the literature review are in line with the AETOS motivation discussed in Sect. 7.2. Robust and self-organised tree overlays depend on the application domain. Most optimisations consider metrics related to physical networks.

It is unclear how other higher-level application-related metrics could influence and change the proposed self-organisation methods. In addition, related work reveals that different applications have different trade-offs. Therefore, combining or weighting multiple optimisation metrics, in an application-independent way, is challenging.

Dynamic protocols, i.e., gossiping protocols, and complementary overlays are effective in many cases. Usually, they are not required to be dedicated for the self-organisation of trees but rather can be reused as existing services in distributed environments. The role of such complementary overlays should be further studied and clarified. The same holds for the proactive or reactive approaches of self-organised systems. Although high proactiveness results in high robustness and resilience to failures, the required cost can be significant with relatively low benefits for the application. Future work should explore the level of proactiveness and reactiveness required for building robust tree overlays for a wide range of applications.

7.4 Approach

The multi-agent systems paradigm, in which individual autonomous agents interact with each other to accomplish their goals, has been successfully applied to management and self-organisation of distributed systems (Brazier et al. 2009; Lopes and Oliveira 1999; Tianfield and Unland 2005). AETOS, a service for building and maintaining on-demand and application-independent robust tree overlays, deploys agents for the purpose of self-organisation.

Overlay hosts (nodes) are the local environment of AETOS agents. These agents act solely within their local environment (and do not migrate).

AETOS agents have (i) *local knowledge*, (ii) *local components* that manage the local knowledge and execute *local tasks*, and (iii) *local layers* of components that create a hierarchy in the information flow. The AETOS service is provided by these agents (and their interaction).

Three local agents participate in AETOS: (i) *the application agent*, (ii) *the self-organisation agent*, and (iii) *the system control agent*. Figure 7.3 illustrates how they interact in the local AETOS environment.

The principle interactions among AETOS agents are outlined as follows:

> *The 'application agent' abstracts the application-specific requirements to application-independent self-organisation requirements by providing a common interface between applications and AETOS. The 'system control agent' turns the self-organisation requirements to self-organisation parameters that the 'self-organisation agent' understands. It then bootstraps, monitors and finally terminates the self-organisation process. Upon termination, the 'self-organisation agent' makes the parent and children neighbours available to the 'application agent' which makes them accessible to the application.*

Note that Fig. 7.3 depicts interaction between the 'system control agent' and other 'system control agents' outside its local environment. Such interaction is optional and beyond the focus of this paper.

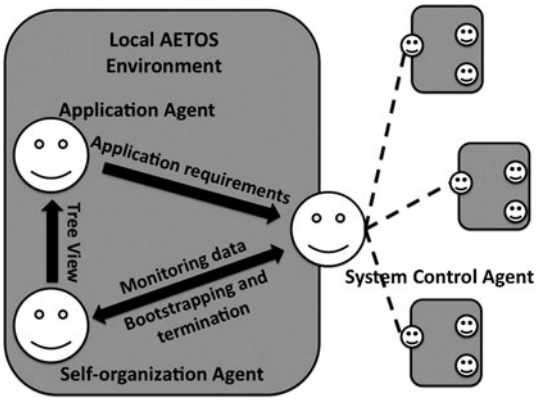


Fig. 7.3 AETOS is based on three agents that interact locally. The 'application agent' provides the application requirements to the 'system control agent'. The latter bootstraps self-organisation, monitors the 'self-organisation agent' and finally terminates self-organisation. When the 'self-organisation agent' is terminated, it makes the parent and the children neighbours available to the 'application agent'

7.5 Application Agent

The 'application agent' provides a generic interface for managing *application requirements* between AETOS and different applications. Note that these requirements are parametrisation settings that make an application work effectively. There is one 'application agent' per application instance. The set of application requirements, denoted by **A**, managed by the 'application agent' are the following:

Robustness (r) This is the abstraction of the optimisation metric on which the self-organisation is based. It can concern any of the previously identified metrics mentioned in Sects. 7.2.2 and 7.3.1. If the application utilises more than one optimisation metrics, the application itself must apply a weighting scheme, or function to derive the abstract robustness r. Robustness is assumed to be a decimal number.

Node Degree (n) The node degree concerns the number of neighbours for each application instance. It denotes the available resources the application reserves for the tree overlay.

Expected Response Time (t_r) This is the time period in which AETOS should return the tree neighbours to the application instance. Higher response times allow better topology optimisations. Section 7.7 explains the use of this parameter by the 'system control agent'.

Note that the above application requirements are the local knowledge of the 'application agent'. The executed tasks are the following:

Register The 'application agent' contacts the 'system control agent' and sends (i) its identifier and (ii) a new tree overlay identifier, to register a new tree overlay in the AETOS service. This information is finally stored in the 'self-organisation agent' together with the reserved space for the tree neighbours.

Build This task concerns the creation and maintenance of a tree overlay. It enables on-demand self-organisation. The 'application agent' sends (i) its identifier, (ii) the tree overlay identifier, and (iii) the set of application requirements **A** to the 'system control agent'. If the utilised tree overlay does not meet the expectations of the application, this task is executed again.

Connect When the set of tree neighbours is received from the self-organisation agent, the set is delivered to the application that finally establishes the connections.

Unregister The 'application agent' contacts the 'system control agent' and sends a tree overlay identifier. The self-organisation for this overlay terminates, and all the information related to this overlay is removed from the 'self-organisation agent'.

By implementing an 'application agent' that incorporates the knowledge and the tasks above, applications have access to the AETOS service.

7.6 Self-Organisation Agent

In AETOS, each node has one local 'self-organisation agent'. The 'self-organisation agent' forms the core of the AETOS system. The self-organisation agent's knowledge, components and 3-layered service architecture are presented in more detail below.

7.6.1 Knowledge

The 'self-organisation agent' has different *partial views* of its distributed environment. A partial view is a list of a finite number of other *node descriptors*. A node descriptor contains information related to the node and its applications, such as its address, connection port, overlay identifier and robustness r. A node descriptor gives the fundamental knowledge which forms the basis for communication between 'self-organisation agents'. The overlay identifier that belongs to a node descriptor received is used by the 'self-organisation agent' to match and extract the respective overlay knowledge that holds locally. Each 'self-organisation agent' has 3 partial views: the *random view*, the *proximity view* and the *tree view*, each described below.

Random View (R) The random view contains the primary knowledge and search space of the 'self-organisation agent'. It consists of a collection of random node descriptors from the distributed environment. Note that the random view is dynamic and changes continuously. This local knowledge creates a global random graph for all overlay hosts. The maintenance and the dynamic changes of the random view are explained in Sect. 7.6.2.

Proximity View (M) The proximity view contains nodes with close proximity to the local node. Proximity is derived by calculating the ranking distance between two nodes. In AETOS, rank values refer to the robustness values r. Therefore, the *robustness distance* between an agent x and an agent y is $d = |r_x - r_y|$. The search space for filling the proximity view is the random view. However, it is also filled by enabling close proximity nodes to exchange neighbours (gossip) and further discover each other faster. Section 7.6.2 illustrates this option. Finally, the proximity view is dynamic and reconfigurable. This means that the ranking function can potentially change by reconfiguring the view appropriately. This aspect is explained in detail in Sect. 7.6.2.

The neighbours of a node in the tree hierarchy are split in two levels, the parent and the children. This concept is applied in the proximity view as well. Two sets of neighbours are defined: (i) the *candidate parents* (**P**) and (ii) the *candidate children* (**C**) such that $\mathbf{M} = \mathbf{P} \cup \mathbf{C}$. Note that the sets are sorted according to robustness r of the node descriptors.

Tree View (T) The tree view is a sorted set with the parent and the children of the local node in the tree overlay. The search space for filling the tree view is the proximity view. The tree view is the one that is provided at the end of self-organisation process to the 'application agent'.

The above views are partial. Their length is a predefined system parameter and depends on the capacity of nodes and on the size of the whole system. For large-scale systems with thousands of nodes, $|\mathbf{R}| \approx 50$ (Jelasity et al. 2007). For the proximity view, a similar scheme is proposed with $|\mathbf{M}| \leq |\mathbf{R}|$. The length of the tree view is $|\mathbf{T}| = n$.

The ratio of the length of the candidate children set over the length of candidate parents set ($\frac{|\mathbf{C}|}{|\mathbf{P}|}$) is proportional to the number of children $c = n - 1$. For example, if $|\mathbf{M}| = 12$ and $c = 3$, then $|\mathbf{C}| = 9$ and $|\mathbf{P}| = 3$. This guarantees that the search space for children and the parent is proportional.

Figure 7.4 illustrates an example of information flow among the views in a self-organisation agent. The proximity of the local random samples from the random view is calculated, and the closest neighbours are inserted in the proximity view. Other close-proximity neighbours are discovered through gossiping. Finally, the candidate neighbours with the highest robustness are acquired for tree neighbours. Upon success, they are inserted in the tree view. Section 7.6.2 provides detailed information about the local interactions and tasks executed by the 'self-organisation agent'.

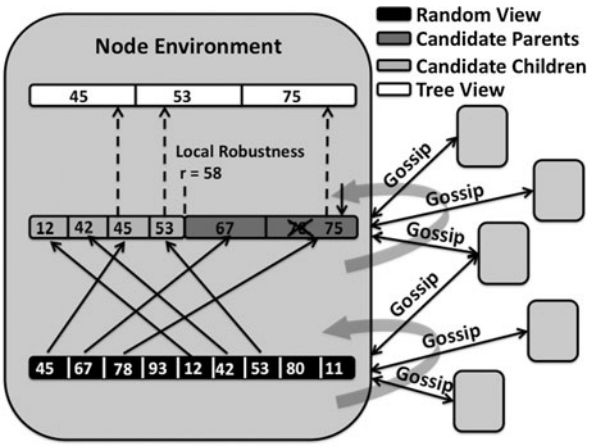


Fig. 7.4 The fundamental knowledge of a 'self-organisation agent' is based on 3 views: (i) the random view, (ii) the proximity view, and (iii) the tree view. The proximity view is filled by random samples and close-proximity neighbours discovered through gossiping. The nodes with the highest robustness in the proximity view are the potential neighbours in the final tree view

7.6.2 Components

The local knowledge and tasks of the 'self-organisation agent' are facilitated in the following components. Figure 7.6 outlines these components and their interactions.

Proximity Manager It holds the proximity view. It interacts with the 'proximity sampling' component and the 'reconfiguration manager' component to update and improve the proximity view. Periodically, it informs the tree manager about the candidate neighbours with the higher robustness r in its proximity view.

Random Sampling This component maintains the random view. This view is updated through a gossiping protocol, that is, the peer sampling service (Jelasity et al. 2007). With the peer sampling service, nodes continuously have random samples of the whole distributed environment and refresh old nodes with new ones. Gossiping creates a dynamic robust overlay on which the tree overlay is based. Readers are referred to Jelasity et al. (2007) for details concerning the peer sampling service.

Proximity Sampling This is the component that realises the gossiping among close-proximity nodes as Fig. 7.4 illustrates. 'Random sampling' discovers close-proximity nodes from random samples. In contrast, 'proximity sampling' further discovers candidate neighbours by exchanging node descriptors between close-proximity nodes. The process of such a gossiping protocol is described in detail in Jelasity et al. (2009). 'Proximity sampling' interacts with the 'proximity manager' to update the proximity view with new candidate parents or children. Note that this component is used to make the system converge faster to the required tree topology.

Reconfiguration Manager The proximity view is not static but rather dynamic and reconfigurable. This means that the ranking function is defined in a dynamic range of robustness values which form a subset of the whole range of values in the

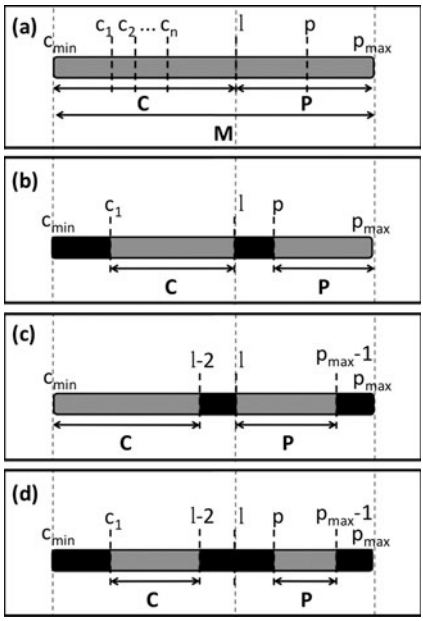


Fig. 7.5 The parent and children candidates in the proximity view. (**a**) Initial proximity view, (**b**) after an upgrade reconfiguration, (**c**) after a downgrade reconfiguration, (**d**) applying an upgrade and a downgrade reconfiguration

proximity view. The 'reconfiguration manager' accesses the 'proximity manager' and is responsible for triggering a number of reconfigurations to the proximity view.

The ranges of robustness values for candidate parents and children are examined below. Let M be the range of the whole set of robustness values that node descriptors contain. All of the indexes refer to robustness values in the proximity view: (i) l points to robustness value of the local node descriptor. (ii) A potential parent p belongs to the candidate parents range P such that $p \in P = [l+1, p_{\max}]$. Similarly, (iii) the potential children $c_1 < c_2 < \cdots < c_n$, with n the number of children, point to the candidate children range C such that $\{c_1, c_2, \ldots, c_n\} \in C = [c_{\min}, l-1]$. Figure 7.5a illustrates the initial ranges of candidate neighbouring sets. The 'reconfiguration manager' can perform the following reconfigurations:

1. *Initialising Configuration*: the ranges of the candidate neighbours are configured as $P = [l+1, p_{\max}]$ and $C = [c_{\min}, l-1]$ respectively. The node descriptor with the higher robustness r in each candidate set is the potential child or parent respectively. In this case, $p = p_{\max}$ and $c_i = l-1$ for the ith potential child.
2. *Upgrade Reconfiguration*: the 'self-organisation agent' has already found a parent or its children, and it seeks to connect with more robust nodes. To achieve this, it binds the starting point of its view to the robustness values of the selected nodes and fills the view with more robust node descriptors. The candidate parents range is reconfigured as $P = [p+1, p_{\max}]$, and the children candidate range as $C = [c_1 + 1, l-1]$. Figure 7.5b depicts the upgrade reconfiguration.
3. *Downgrade Reconfiguration*: if a previously selected candidate neighbour has rejected the connection, the view is updated with less robust nodes. In this case, the candidate ranges are updated as $P = [l+1, p_{\max} - 1]$ and $C = [c_{\min}, l-2]$

respectively. Figure 7.5c illustrates how the view is updated in this case. Note that the downgrade reconfiguration is performed step-by-step, decrementing the positions by one for every rejected parent or child connection respectively.

The 'reconfiguration manager' has the option to switch from a downgrade or upgrade configuration back to the initial one. Furthermore, the proximity view can be a result of both an upgrade and a downgrade reconfiguration. Figure 7.5d illustrates an example of this case. Any applied reconfiguration keeps the length of the proximity view equal or lower than the initial maximum length.

Tree Manager The Tree Manager manages the connectivity of the tree overlay and interacts with other nodes to establish the parent and children connections. The interactions are based on the exchange of 4 messages: (i) the *request* of a parent or child connection, (ii) the *acknowledgement* of a request, (iii) the *rejection* of a request, and (iv) the *removal* of a parent or child connection.

In its active state, the 'tree manager' periodically accesses the 'proximity manager' and receives the candidate parent and child with the highest robustness r. It sends a 'parent and child request' to each of them respectively. If the 'proximity manager' cannot provide candidate neighbours to the 'tree manager' for a prespecified period of time, it reports this information to the 'reaction manager'.

The passive state of the 'tree manager' defines the appropriate reactions to the messages received. For a 'parent or child request', the reactions are the following:

1. It checks if the robustness r of the two communicating nodes are consistent. This means that the value of the parent should be higher than the value of the child. If inconsistencies occur due to changes in the values of robustness, the 'tree manager' sends a 'rejection' message to the requesting agent with information about the value of local robustness.
2. If there are no inconsistencies, the 'tree manager' either
   a. updates and inserts the node that sent the 'parent/child request' in its tree view. In this case, the 'tree manager' replies with an 'acknowledgement'. If the update of the tree view is performed by replacing an existing node descriptor with one with higher robustness, then a 'removal' message is sent to the replaced node. Or,
   b. it rejects the request and a 'rejection' message is sent. In this case, the existing parent or children are more robust than the node that sent the request.

   In both cases the reply-messages contain information that reflects the more recent values of the robustness r.
3. A report is sent to the local 'reaction manager'.

If the 'tree manager' receives an 'acknowledgement' of its request, it performs:

1. An update of its tree view by inserting the new neighbour. If the update is a replacement, it sends a 'removal' messages to the replaced node.
2. A report to the local 'reaction manager'.

The 'rejection' message triggers the following:

1. A report to the local 'reaction manager'.

Finally, in case of a 'removal' message, 'tree manager' performs:

1. Removal of the parent or one of the children.
2. A report to the local 'reaction manager'.

These messages form the basic interactions among the AETOS agents to configure the tree overlay connections.

Reaction Manager It receives reports from the 'tree manager' concerning the configuration of the tree connections. Based on these reports, it triggers the appropriate reconfigurations in the 'reconfiguration manager'.

An upgrade reconfiguration is triggered when a new parent or the last child is added in the candidate parents or children respectively. A downgrade reconfiguration is applied when a 'parent of child request' is rejected or a removal is performed in a parent or child. Finally, the initialising reconfiguration is performed before the upgrade or downgrade reconfigurations to overwrite the old ones.

7.6.3 Service Layer Architecture

The interactions of the components in the 'self-organisation agent' can be outlined in the following 3-layer hierarchy:

PAROS The *ProActive Robust Overlay Sampling* is the underlying overlay that provides high robustness in the tree overlay. It guarantees that the network remains connected, and it is not clustered due to node departures or failures.

ARMOS The *Adaptive Rank-based Middleware Overlay Service* is a proximity-driven reconfigurable overlay. It incorporates the 'tree manager', the 'proximity sampling' and the 'reconfiguration manager'. It is based on PAROS and supports the connectivity of the tree overlay by providing candidate neighbours.

ATOM The *Adaptive Tree Overlay Management* is responsible for configuring the tree connections and provides feedback to ARMOS for improving the candidate neighbours.

Figure 7.6 outlines the 3-layer hierarchy and the components of the 'self-organisation agent'. The sequence of interactions is as follows: (1) 'random sampling' provides periodically random samples to the 'proximity manager'. From these random samples, the ones with close proximity are selected and stored in the proximity view. (2) 'proximity sampling' exchanges node descriptors with close proximity nodes for improving the proximity view. (3) Periodically, the 'proximity manager provides the best candidate neighbours to the 'tree manager'. The latter interacts with these candidates to establish tree connections. (4) The result of these interactions is reported to the 'reaction manager' (5) that triggers the appropriate reconfigurations in the 'reconfiguration manager'. (6) Finally, the proximity view is reconfigured and new candidate neighbours can be provided to the 'tree manager'.

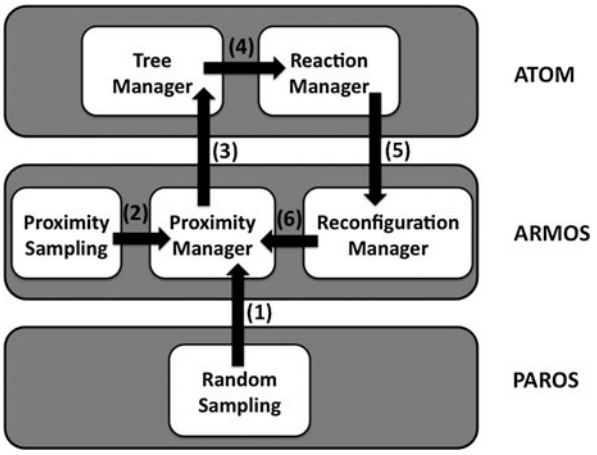


Fig. 7.6 The 3-layer hierarchical interactions of the components within the 'self-organisation agent'. The numbers denote the sequence of interactions between the components. The *arrows* (3), (4), (5) and (6) depict the feedback loop which forms the core of adaptivity in AETOS

Note that the feedback loop between the 'proximity manager', 'tree manager', 'reaction manager' and 'reconfiguration manager' forms the core of adaptivity in AETOS.

7.7 System Control Agent

The 'system control agent' acts as a proxy between the 'application agent' and 'self-organisation agent'. It keeps information about the registered overlays and provides this information to the 'self-organisation agent'. It also receives the application requirements for each overlay and monitors the self-organisation process. With this information, it can control locally the bootstrapping and termination of the self-organisation.

Bootstrapping The 'system control agent' initially guarantees that the robustness values are unique. This is achieved by assigning a unique comparable random number in the robustness value r. It then feeds the robustness r and the number of children $c = n - 1$ to the 'self-organisation agent'. Therefore, the 'self-organisation agent' is able to start executing its component tasks.

Termination Termination is based on the expected response time t_r. The 'system control agent' monitors the 'self-organisation agent'. When the runtime exceeds the t_r, it terminates the self-organisation.

At this moment of local convergence, the agent (i) stops the participation of the agent in the self-organisation process and (ii) enables the 'tree manager' to provide the tree view to the application. Note that the node can be still contacted when is not participating in the self-organisation. In this case, it notifies the node about its current terminated state.

In this termination approach, the application is the one that defines, through its requirements, when the self-organisation terminates rather than the underlying AETOS system. The motivation for this decision is that the stability of the tree overlay is evaluated with respect to the application requirements, and thus it must be the one that influences the termination of the self-organisation.

7.8 Evaluation of the Proposed Approach

AETOS is implemented and evaluated in ProtoPeer (Galuba et al. 2009), an asynchronous simulation platform for large-scale distributed systems. ProtoPeer provides a generic interface for enabling the step from single-machine to multiple-machine simulation and finally to live deployment.

This section focuses on the evaluation of the 'self-organisation agent'. The goal of the evaluation is to reveal the cost-effectiveness of AETOS in the connectivity of two different tree topologies. In this section, connectivity *refers to the percentage of the total number of nodes connected to the main tree. The convergence of connectivity is investigated under varying length of the random view and two different network sizes.*

The input settings in the ProtoPeer simulation environment represent the 'application agent'. The 'self-organisation agent' is implemented as three services or 'peerlets' in ProtoPeer terminology. Each service corresponds to a layer in the architecture of Fig. 7.6. In the first layer, the peer sampling service (Jelasity et al. 2007) is the implementation of the 'random sampling' component. In the middle layer, the 'proximity manager' and the 'reconfiguration manager' are implemented. The implementation of 'proximity sampling' is part of ongoing work and is not part of AETOS in the results illustrated in this section. However, the implications of this missing component are discussed in this section. The two components of the ATOM layer, the 'tree manager' and 'reaction manager', are facilitated in a peerlet of the 'self-organisation agent'. Finally, the evaluation of the bootstrapping and termination by the 'system control agent' is part of future work.

7.8.1 Simulation Settings

Three group of experiments are performed in two different simulation environments. Table 7.1 outlines the simulation parametrisation in these two environments. 'Simulation environment 1' has 121 nodes. 'Simulation environment 2', a larger-scale network, has 1093 nodes. The first two groups of experiments run for 2500 iterations, and the third for 400. The latter group of experiments runs for fewer iterations due to restrictive memory scalability of the ProtoPeer measurement infrastructure. The ProtoPeer environment supports bootstrapping of the system in a ring topology in the first 6 iterations from which the peer sampling service and the components in the higher levels are initialised.

Table 7.1 Simulation environments

Parameter	Simulation environment 1	Simulation environment 2
Number of nodes (**N**)	121	1093
Number of children (c)	3–5	3–5
View selection policy	swapper	swapper
Random view length ($\|\mathbf{R}\|$)	4–20	40
Candidate parents length ($\|\mathbf{P}\|$)	2	3
Candidate children length ($\|\mathbf{C}\|$)	4	5
Number of iterations	2500	400
Requests frequency	2 per iteration	2 per iteration

The *swapper* selection policy used within the peer sampling service (Jelasity et al. 2007) is used to increase randomness in the local node samples. In 'simulation environment 1' the length of the random view **R** is varied between 4-20. In 'simulation environment 2' the length of the random view **R** is fixed to 40. The length of the view of candidate parents is chosen to be smaller than the view of the candidate children and is $|\mathbf{P}| = 2$, $|\mathbf{P}| = 4$ for the first and $|\mathbf{P}| = 3$, $|\mathbf{P}| = 5$ for the second simulation environment.

Finally, nodes are organised in two tree topologies: (1) a tree for which the number of children to which the 'application agents' try to connect is 3 and (2) a tree for which the number of children to which the 'application agents' try to connect is 5. As a result with a fixed number of nodes, the trees have different number of levels. The robustness r assigned to the 'self-organisation agent' is a unique random number between 0 and 100. Note that in every iteration the 'self-organisation agent' potentially sends one parent and one child request, thus the frequency of requests is 2 per iteration.

7.8.2 Results

The first group of experiments runs in 'simulation environment 1' in which the number of children to which agents aim to connect is equal to 3. Figure 7.7(a) illustrates the convergence of connectivity by varying the length of the random view. Figure 7.7(b)–(e) depicts the communication cost of AETOS expressed in the number of messages generated by the ATOM layer of the 'self-organisation agent'.

The second group of experiments also runs in 'simulation environment 1', but in this case the number of children to which agents aim to connect is equal to 5. Figure 7.8(a) illustrates the convergence of the connectivity by varying the length of the random view. Figure 7.8(b)–(e) depicts the communication cost of AETOS expressed in the number of messages generated by the ATOM layer of the 'self-organisation agent'.

Finally, the last group of experiments runs in 'simulation environment 2' for $c = 3$ and $c = 5$. Figure 7.9 illustrates the connectivity convergence in this settings.

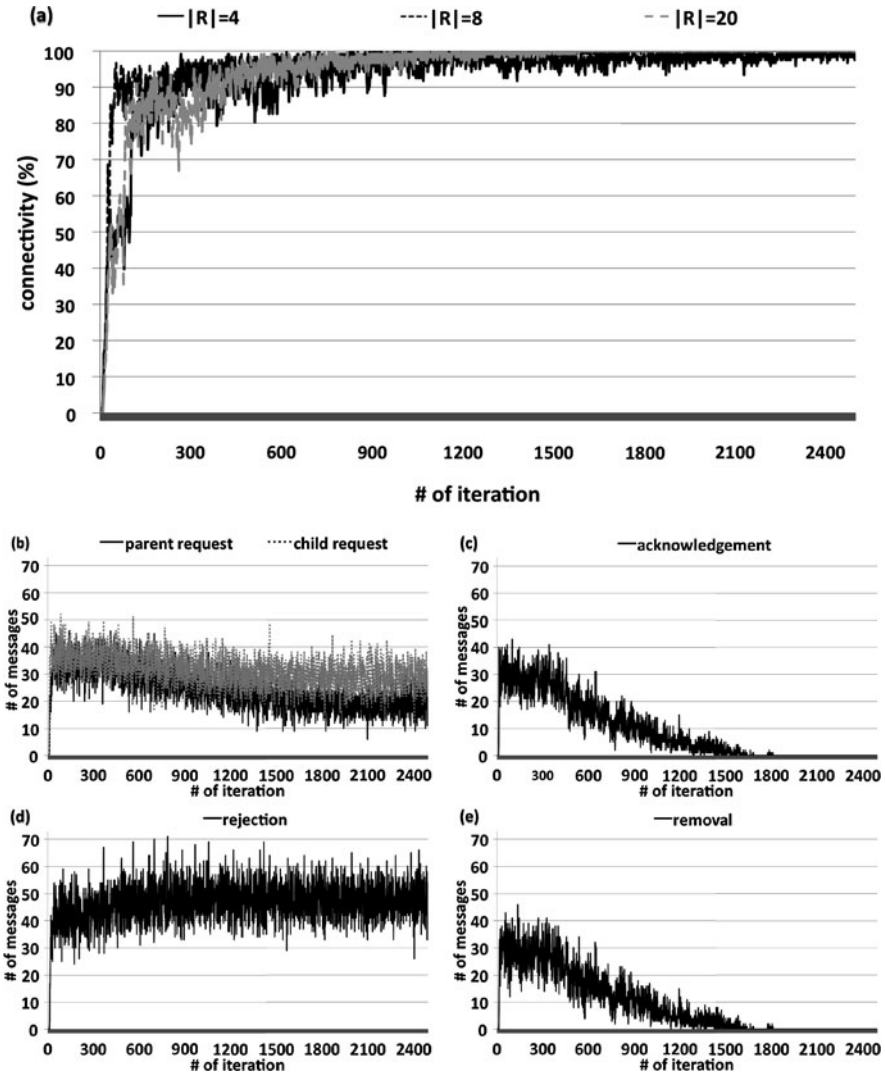


Fig. 7.7 Cost-effectiveness of AETOS in 'simulation environment 1' for $c = 3$. (**a**) Connectivity convergence for different length of random views. (**b**)–(**e**) Number of messages generated by the ATOM layer of the 'self-organisation agent' for $|\mathbf{R}| = 20$

In summary, the above results show that AETOS can achieve a high degree of connectivity in both simulation environments. AETOS converges to 90% connectivity in less than 150 iterations. An exception is the case of 'simulation environment 2' with $c = 3$, in which connectivity approaches 60% in the 400th iteration. Section 7.8.3 explains in detail the behaviour of AETOS in these simulation experiments.

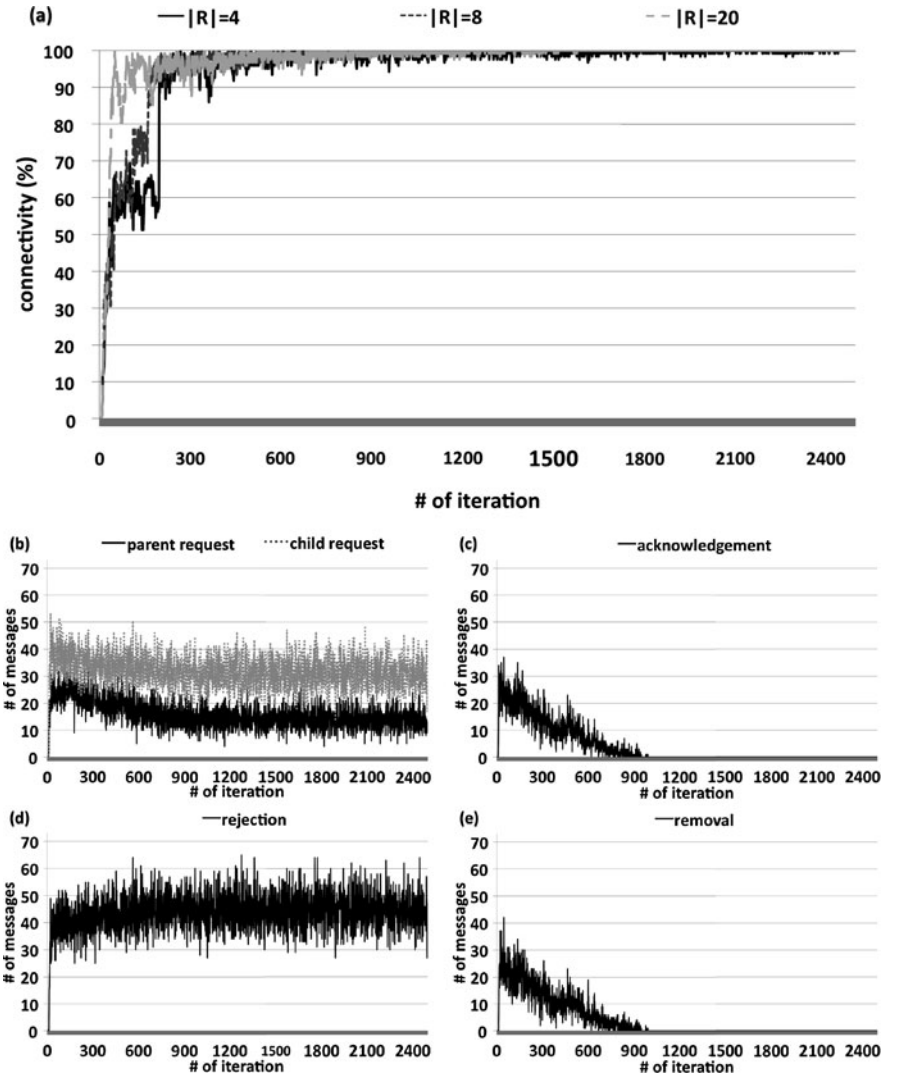


Fig. 7.8 Cost-effectiveness of AETOS in 'simulation environment 1' for $c = 5$. (**a**) Connectivity convergence for different lengths of random views. (**b**)–(**e**) Number of messages generated by the ATOM layer of the 'self-organisation agent' for $|\mathbf{R}| = 20$

7.8.3 Discussion of Experimental Results

The results reveal that a certain percentage of connectivity can be achieved within relatively few iterations. For example, 50% connectivity can be achieved in less than 100 iterations in 'simulation environment 1' and between 150–400 iterations in 'simulation environment 2'. In contrast, for connectivity higher than 98%, AETOS convergence lasts much longer, requiring 436 iterations in 'simulation envi-

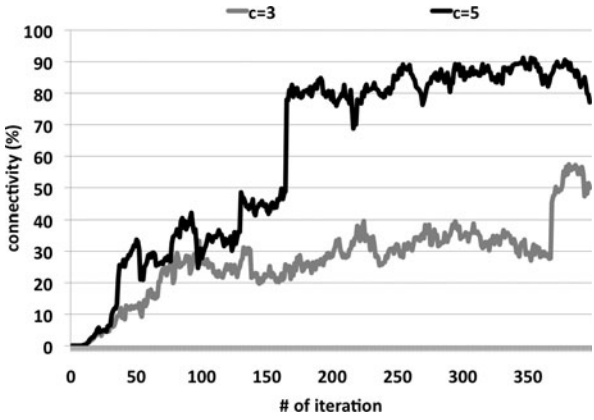


Fig. 7.9 Connectivity convergence in 'simulation environment 2' for $c = 3$ and $c = 5$

ronment 1'. In this environment, increasing connectivity from 90% to 98% requires more than 250 additional iterations. This effect is more significant in 'simulation environment 2' in which connectivity seems to converge 10%–30% more slowly than 'simulation environment 1'. The peer sampling service provides a bounded random search space, and thus convergence speed decreases as the size of the network or the topology complexity increases. The connectivity jump from 50% to about 80% in the 150th iteration in Fig. 7.9 is explained by the connection of a large branch of nodes to the main body of the tree.

The communication cost of the ATOM layer is related to three things: (i) request frequency, (ii) convergence of the system, and (iii) effectiveness in the termination of self-organisation. The parent and child requests decrease during convergence 40%–45% and 25%–35% respectively. This is caused by the effect of the reconfigurations and the increase in the tree connectivity. In contrast, rejections increase 25%–30% as there are more nodes already connected that can potentially reject requests. After convergence, the number of messages is stabilised. At this point the system can be terminated and thus, alleviate the network from this constant communication overhead. Removal and acknowledgement messages decrease proportionally to the convergence time. This is expected, as nodes in a tree with 100% connectivity do not perform any removals or acknowledgements. Note that the communication cost of the PAROS layer is constant and dependent on the network size and the gossiping period.

An increase in the length of random views makes connectivity convergence faster in both simulation environments. This can be explained by the better global knowledge that the 'self-organisation agents' have of the system. Therefore, they can improve the quality of the candidate neighbours select to which they potentially connect. Finally, the number of children c influences the cost-effectiveness of AETOS significantly. A different number of children results in different topologies. In each simulation environment setting of this section, trees have the same network size with a different number of levels. Connectivity increases from 81% ($c = 3$) to 97% ($c = 5$) in 'simulation environment 1' and from 50% ($c = 3$) to 77% ($c = 5$) in 'sim-

ulation environment 2' at the 400th iteration. Furthermore, communication cost also decreases 17% by increasing c in 'simulation environment 1'.

The future addition of 'proximity sampling' is expected to enhance the quality of the proximity view. More specifically, it is expected to (i) decrease the connectivity convergence time as the self-organisation agent will update the proximity view faster after the performed reconfigurations and (ii) decrease the communication cost of the ATOM layer as it is related to convergence time.

7.9 Conclusions and Future Work

This chapter proposes AETOS, the Adaptive Epidemic Tree overlay Service. AETOS is an agent-based system that builds and maintains application-independent tree overlays, on demand. To this purpose three local agents are defined to (i) abstract application requirements to self-organisation requirements, (ii) self-organise nodes in various optimised tree topologies based on these requirements, and (iii) control the bootstrapping and termination of self-organisation. Experiments show that a high level of connectivity can be acquired and that cost-effectiveness of self-organisation is highly correlated to the available local knowledge, the tree topology and the network size.

These results are promising. Further research will include extension of the current system with a 'proximity sampling' component, study the effects of a distributed simulation environment and application of AETOS in a more realistic domain.

Acknowledgements The authors are grateful to the NLnet Foundation and Delft University of Technology for their support. http//www.nlnet.nl.

References

B. Akbari, H. R. Rabiee, and M. Ghanbari. DPOCS: A dynamic proxy architecture for video streaming based on overlay networks. In *IEEE MICC & ICON '05*, volume 1, page 6, November 2005.

G. An, D. Gui-guang, D. Qiong-hai, and L. Chuang. BulkTree: An overlay network architecture for live media streaming. *Journal of Zhejiang University*, 7(1):125–130, 2006.

S. Banerjee, C. Kommareddy, K. Kar, S. Bhattacharjee, and S. Khuller. Construction of an efficient overlay multicast infrastructure for real-time applications. In *INFOCOM*, volume 2, pages 1521–1531, 2003.

S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan. Resilient multicast using overlays. *IEEE/ACM Transactions on Networking*, 14(2):237–248, 2006.

R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In *IPTPS*, pages 256–267, 2003.

S. Birrer and F. E. Bustamante. A comparison of resilient overlay multicast approaches. *IEEE Journal on Selected Areas in Communications*, 25(9):1695–1705, 2007.

F. M. T. Brazier, J. O. Kephart, M. Huhns, and H. Van Dyke Parunak. Agents and service-oriented computing for autonomic computing: A research agenda. *IEEE Internet Computing*, 13(3):82–87, May 2009.

A. J. Chakravarti, G. Baumgartner, and M. Lauria. The organic grid: self-organizing computation on a peer-to-peer network. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(3):373–384, 2005.

J. A. Chaudhry and S. Park. Ahsen—autonomic healing-based self management engine for network management in hybrid networks. In *GPC*, pages 193–203, 2007.

P. Costa and D. Frey. Publish–subscribe tree maintenance over a DHT. In *ICDCSW '05: Proceedings of the Fourth International Workshop on Distributed Event-Based Systems (DEBS) (ICDCSW'05)*, pages 414–420, Washington, 2005. IEEE Computer Society.

C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14(1):78–88, 2000.

D. England, B. Veeravalli, and J. B. Weissman. A robust spanning tree topology for data collection and dissemination in distributed environments. *IEEE Transactions on Parallel and Distributed Systems*, 18(5):608–620, 2007.

Z. Fei and M. Yang. A proactive tree recovery mechanism for resilient overlay multicast. *IEEE/ACM Transactions on Networking*, 15(1):173–186, 2007.

D. Frey and A. L. Murphy. Failure-tolerant overlay trees for large-scale dynamic networks. In *P2P '08: Proceedings of the 2008 Eighth International Conference on Peer-to-Peer Computing*, pages 351–361, Washington, 2008. IEEE Computer Society.

W. Galuba, K. Aberer, Z. Despotovic, and W. Kellerer. ProtoPeer: a P2P toolkit bridging the gap between simulation and live deployment. In *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–9, ICST, Brussels, Belgium, 2009.

A. González-Beltrán, P. Milligan, and P. Sage. Range queries over skip tree graphs. *Computer Communications*, 31(2):358–374, 2008.

H. V. Jagadish, B. C. Ooi, K.-L. Tan, Q. H. Vu, and R. Zhang. Speeding up search in peer-to-peer networks with a multi-way tree structure. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, 2006. ACM.

M. Jelasity, A. Montresor, and O. Babaoglu. T-man: Gossip-based fast overlay topology construction. *Computer Networks*, 53(13):2321–2339, 2009.

C. Y. Lee and H. Dong Kim. Reliable overlay multicast trees for private Internet broadcasting with multiple sessions. *Computers & Operations Research*, 34(9):2849–2864, 2007.

J. Leitao, J. Pereira, and L. Rodrigues. Epidemic broadcast trees. In *SRDS '07: Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems*, pages 301–310, Washington, 2007. IEEE Computer Society.

Y. Li and W. T. Ooi. Distributed construction of resource-efficient overlay tree by approximating MST. In *ICME*, pages 1507–1510, 2004.

M. Li, W.-C. Lee, and A. Sivasubramaniam. DPTree: A balanced tree based indexing framework for peer-to-peer systems. In *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 12–21, Washington, 2006. IEEE Computer Society.

J. Liu and M. Zhou. Tree-assisted gossiping for overlay video distribution. *Multimedia Tools and Applications*, 29(3):211–232, 2006.

Y. Liu, Y. Guo, and C. Liang. A survey on peer-to-peer video streaming systems. *Peer-to-Peer Networking and Applications*, 1(1):18–28, 2008.

R. P. Lopes and J. L. Oliveira. Software agents in network management. In *ICEIS*, pages 674–681, 1999.

M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. Gossip-based peer sampling. *ACM Transactions on Computer Systems*, 25(3):8, 2007.

P. Merz and S. Wolf. TreeOpt: Self-organizing, evolving P2P overlay topologies based on spanning trees. In *SAKS'07*, Bern, Switzerland, 2007.

E. Pournaras, G. Exarchakos, and N. Antonopoulos. Load-driven neighbourhood reconfiguration of Gnutella overlay. *Computer Communications*, 31(13):3030–3039, 2008.

E. Pournaras, M. Warnier, and F. M. T. Brazier. A distributed agent-based approach to stabilization of global resource utilization. In *Proceedings of International Conference of Complex Intelligent and Software Intensive Systems (CISIS'09)*, March 2009a.

E. Pournaras, M. Warnier, and F. M. T. Brazier. Adaptive agent-based self-organization for robust hierarchical topologies. In *ICAIS '09: Proceedings of the International Conference on Adaptive and Intelligent Systems*, IEEE, New York, 2009b.

W. Pugh. Skip lists: a probabilistic alternative to balanced trees. *Communications of the ACM*, 33(6):668–676, 1990.

G. Tan, S. A. Jarvis, X. Chen, and D. P. Spooner. Performance analysis and improvement of overlay construction for peer-to-peer live streaming. *Simulation*, 82(2):93–106, 2006.

S.-W. Tan, G. Waters, and J. Crawford. MeshTree: Reliable low delay degree-bounded multicast overlays. *International Conference on Parallel and Distributed Systems*, 2:565–569, 2005.

C. Tang and C. Ward. GoCast: Gossip-enhanced overlay multicast for fast and dependable group communication. In *DSN '05: Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pages 140–149, Washington, 2005. IEEE Computer Society.

H. Tianfield and R. Unland. Towards self-organization in multi-agent systems and grid computing. *Multiagent Grid Systems*, 1(2):89–95, 2005.

F. Wang, Y. Xiong, and J. Liu. mTreebone: A hybrid tree/mesh overlay for application-layer live video multicast. In *IEEE ICDCS*, page 49, 2007.

H. Zhuge and L. Feng. Distributed suffix tree overlay for peer-to-peer search. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):276–285, 2008.

Chapter 8
Filtering Order Adaptation Based on Attractor Selection for Data Broadcasting System

**Shinya Kitajima, Takahiro Hara,
Tsutomu Terada, and Shojiro Nishio**

Summary Recent spread of different data broadcasting services leads to provide enormous and various heterogeneous data. Since data that a client needs are a part of them, there has been an increasing interest in information filtering techniques where a client automatically chooses and stores the necessary data. Generally, when a client performs filtering, it applies some filters sequentially, and the time required for filtering changes according to the order of filters. On the other hand, in recent years, there have been many studies about attractor selection which is an autonomous parameter control technique based on the knowledge from living organisms. In this chapter, in order to reduce the load for filtering, we propose novel methods which adaptively change the order of filters according to the change in broadcast contents. These methods adaptively decide the control parameters for filtering by using attractor selection.

8.1 Introduction

Recent spread of different data broadcasting services leads to provide enormous and various heterogeneous data. In a broadcast system, while the server can

S. Kitajima (✉) · T. Hara · S. Nishio
Dept. of Multimedia Eng., Grad. School of Information Science and Technology,
Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
e-mail: lastis@infoseek.jp

T. Hara
e-mail: hara@ist.osaka-u.ac.jp

S. Nishio
e-mail: nishio@ist.osaka-u.ac.jp

T. Terada
Dept. of Electrical and Electronics Eng., Grad. School of Science and Technology, Kobe
University, 1-1 Rokkodai, Nada, Kobe 657-8501, Japan
e-mail: tsutomu@eedept.kobe-u.ac.jp

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_8,

broadcast large amount of data at a time, clients are typically mobile terminals whose storage are limited (Belkin and Croft 1992; Bell and Moffat 1996; Sawai et al. 2004). Therefore, there has been an increasing interest in information filtering techniques which automatically choose and store necessary data on the client's storage.

Generally, when a client performs filtering, it applies some filters sequentially. The time required for filtering changes according to the order of filters, since the number of data items that match each filter and the filtering load are different among filters. When the filtering load is high, the filtering speed might become slower than the receiving speed of data. Thus, in an information filtering system, how to determine the order of filters is a crucial problem.

On the other hand, in recent years, there have been several studies about *attractor selection* which is an autonomous parameter control technique based on the knowledge from living organisms (Kashiwagi et al. 2006; Leibnitz et al. 2005, 2009). By using attractor selection, the system can control parameters depending on the situation autonomously, and thus it can cope with changes of the system environment flexibly.

In this chapter, in order to reduce the load for filtering process, we propose novel methods which adaptively change the order of filters adapting to the change in broadcast contents (Kitajima et al. 2009). These methods adaptively decide the control parameters for filtering by using attractor selection. Furthermore, we show the results of simulation experiments, from which we confirm that the proposal methods improve the load for filtering process compared with other methods.

The remainder of this chapter is organized as follows. Section 8.2 describes the outline of an information filtering system, and Sect. 8.3 introduces attractor selection. Section 8.4 explains our proposed methods in details. Section 8.5 evaluates the performance of our methods. Finally, we conclude the chapter in Sect. 8.6.

8.2 Information Filtering System

8.2.1 Mobile Environment

There are several data broadcasting services that have been already available, e.g., those using a surplus band of terrestrial broadcasting, news distributions on the Internet, and bidirectional data services using satellite broadcasting. In such data broadcasting services, the server can send enormous information to a large number of users at a time. However, the data wanted by a user are generally just a small part of the broadcast data.

In this chapter, we assume an urban data broadcasting service in town which is thought to be common in the near future. Figure 8.1 shows a system environment assumed in this chapter. In this environment, mobile users equipped with portable devices such as PDAs and smartphones (mobile clients) walk in town and receive broadcast data via the wireless channel from the nearest server. Some conventional

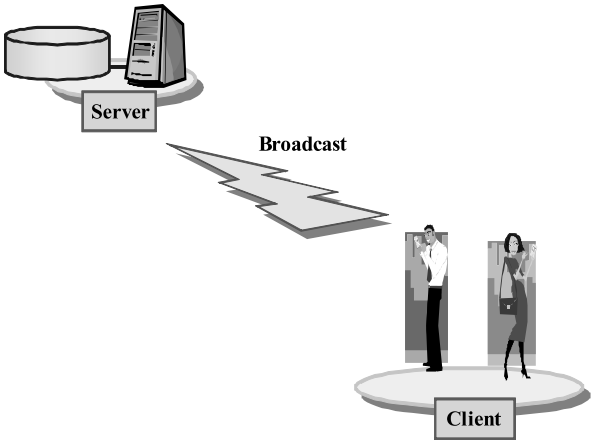


Fig. 8.1 Assumed environment

works such as (Acharya et al. 1995) also assume such an information broadcasting system.

Broadcast contents are mainly text data, and the data of various genres such as real time information like news and weather forecast, local store information, and event information are broadcast. Since mobile clients have a limit for the storage, information filtering to automatically choose the necessary information for users is highly required. We call such a system as an *information filtering system*.

8.2.2 Filtering Architecture

In the information filtering system shown in Fig. 8.2, each client stores broadcast data items once into its receiving buffer, then performs filtering operations when the number of the received items reaches the predetermined constant, and stores only the necessary data items on the storage. Here, we show an example of filtering broadcast data. If a user wants to get data on today's news about sports, the system performs filtering operations by using three kinds of filters to get contents whose (i) category is "news," (ii) issue date is today, and (iii) topic is "sports."

There are various kinds of filters, e.g., a filter which gets items that match specified category or keyword, a filter which gets items that are given a timestamp of the particular period of time, a filter which gets items of high relevance by using the cosine correlation between the user's preference and items (Salton and McGill 1983). The load of applying these filters is different with each other. For instance, the load of calculating the cosine correlation is heavier than that of simple keyword matching.

Moreover, some filters are often applied at the same time as shown in the above example. If the filters do not include ranking operations and do include only se-

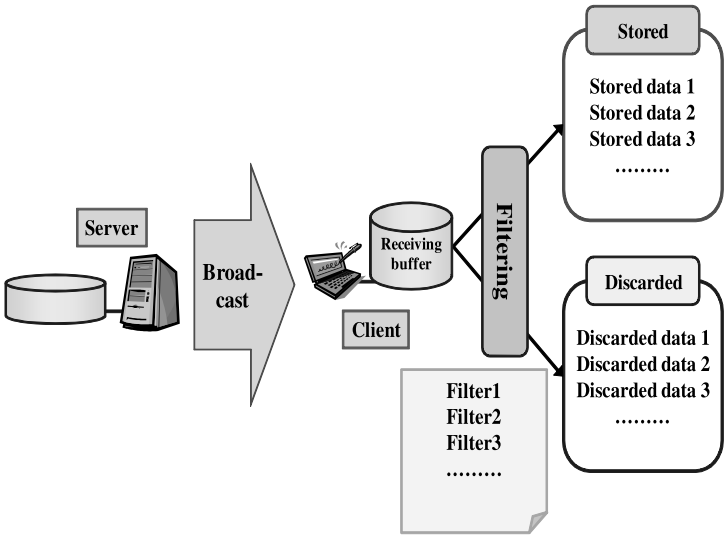


Fig. 8.2 Information filtering system

lection operations, the order of applying filters does not affect the result of filtering (Sawai et al. 2004).

8.2.3 Filtering Cost

If there are multiple filters to apply, the order of applying filters influences the *filtering cost* since the number of data items that match each filter and the *processing cost* of the filter are different among filters. Here, the filtering cost represents the time to perform the filtering operations as a numerical value, and the processing cost means the processing time of the filter per data item. The time to apply a filter is proportional to the processing time of the filter per data item and the number of data items that are applied the filter.

Figure 8.3 shows an example that the filtering cost changes according to the order of filters. In this figure, Tables (a) and (b) show a case in which there are same five broadcast data items stored in the receiving buffer of a client, but the order of applying filters is different. Here, let us assume that two attributes are attached to each data item (Attribute1: category, Attribute2: keyword) that represent the contents of the item. For instance, item 1 belongs to category "news," and its keyword is "sports."

In Table (a), a filter to select data items in category "news" is applied to five items at first, and then another filter to select data items with keyword "sports" is applied to two items that are selected by the first filter. Let us also assume the processing cost of both filters is 1. In this case, the total cost becomes 7. On the other hand, in Table (b), the keyword filter ("sports") is applied to the five items at first, and then

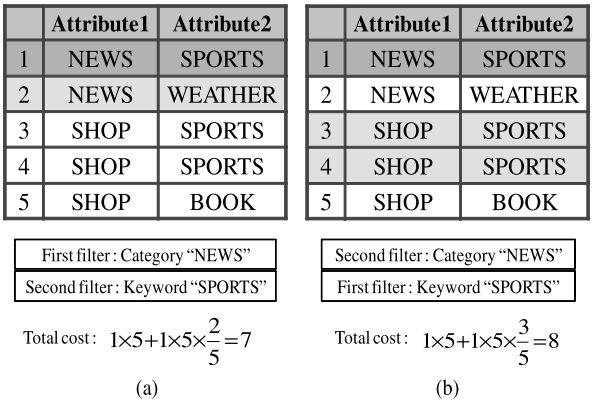


Fig. 8.3 Calculation of filtering cost

the category filter ("news") is applied to three items that are selected by the keyword filter. In this case, the total cost becomes 8. In this way, the filtering cost changes according to the number of data items that match each filter and the processing cost of the filter.

If the processing speed of the filters becomes lower than the receiving speed, the receiving buffer overflows since data are continuously broadcast. In addition, users use their mobile terminals not only to receive the broadcast data but also for other services such as a navigation tool and Video on Demand (VOD). Therefore, the filtering cost should be as small as possible.

In our assumed environment, there are various broadcast data whose contents dynamically change, such as real time information like news, weather, and local store and event information. Moreover, users' demand also dynamically changes. In such an environment, a method that can adaptively and dynamically decide the order of filters is needed.

If the time to apply filters to data items is longer than the time to receive these data items, it is impossible to apply filters to all data items. Therefore, enough processing speed is required to perform filtering. Moreover, enough storage is required since each client stores broadcast data items once into its receiving buffer. The required processing speed changes according to the bandwidth of the broadcast channel and the size of a data item, and the required storage changes according to the size of the receiving buffer and the size of a data item.

8.3 Attractor Selection

8.3.1 Adaptive Response by Attractor Selection

In this subsection, we describe the outline of the attractor selection mechanism, which has been proposed in Kashiwagi et al. (2006). The authors claim that or-

ganisms form a complicated network system having the networks of many hierarchies such as gene, protein, and metabolism, which they call the *organism networks*. When different organism networks meet together, they reach to a stable state (*attractor*) while changing their structure and route and form an organism symbiosis network. This mechanism has many properties such as expansibility, autonomy, toughness, flexibility, adaptability, and variety, which are also needed in an information network and system. Here, "symbiosis" means that multiple different organisms interact with each other and live by supplying the properties to others which they do not have mutually.

It is necessary to adapt to a new environment flexibly while two kinds of organisms without having met before process to form symbiosis relations. However, they have not experienced this environment change in the past, so that they cannot prepare for a hereditary program corresponding to it.

By conventional studies, it becomes clear that transition from an original stable state to a new stable state by the reorganization of the gene metabolism network (three classes of networks of gene, protein, and metabolism), and interaction between the cells by the chemical substance are important. Based on this, the authors suggest a new mechanism called the adaptive response by attractor selection.

In Kashiwagi et al. (2006), to represent a complicated gene metabolism network simply, a model having double feedback loops is defined as follows:

$$\frac{dm_1}{dt} = \frac{syn(act)}{1+m_2^2} - deg(act)\cdot m_1 + \eta_1, \tag{8.1}$$

$$\frac{dm_2}{dt} = \frac{syn(act)}{1+m_1^2} - deg(act)\cdot m_2 + \eta_2, \tag{8.2}$$

$$syn(act) = \frac{6act}{2+act}, \tag{8.3}$$

$$deg(act) = act. \tag{8.4}$$

Here, m_1 and m_2 are mRNA densities made by operons 1 and 2, where an operon is one of the functional units existing on a genome; η_1 and η_2 in the third term on the right side in (8.1) and (8.2) are noises. Equations (8.1)–(8.4) show that when the activity *act* is high, the mRNA density rarely changes. This is because the first term on the right side in (8.1) and (8.2) becomes dominant. On the other hand, when the activity is low, the third term in (8.1) and (8.2), i.e., the noise, becomes dominant, and the system tries to transit another stable state. The activity *act* changes according to the following equation:

$$\frac{dact}{dt} = \frac{pro}{((\frac{Nut\_th_1}{m_1+Nut_1})^{n_1}+1)\times((\frac{Nut\_th_2}{m_2+Nut_2})^{n_2}+1)} - cons\times act. \tag{8.5}$$

Here, Nut_1 and Nut_2 are the supply densities of nourishment from the outside for operons 1 and 2; and Nut_th_1 and Nut_th_2 are their thresholds. *pro* and *cons* are the coefficients of production and consumption of the activity, and n_1 and n_2 are appropriate constant numbers.

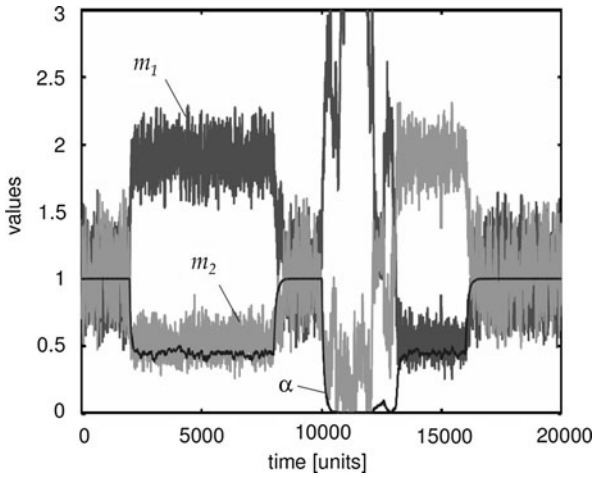


Fig. 8.4 Response of the double feedback loop

Figure 8.4 shows the responses of these double feedback loops. It can be seen that when the outside supply for one nourishment is cut, the attractor making up for the lack is selected. There are two absorption domains in this environment, but only an appropriate attractor is selected. This is because the fluctuation by noises becomes large when the environment becomes worse and activity *act* becomes low, and then, the system is absorbed by that attractor while it approaches to the attractor and recovers the activity. This behavior is an environmental adaptation by attractor selection.

8.3.2 Advantages of Attractor Selection

The fully premeditated construction of systems has become impossible with the rapid large-scaling and complexifying of recent information systems. Therefore, in recent years, system management techniques to adapt for changes of the environment flexibly and autonomously have become crucial.

For instance, in the field of network design, conventional systems have been fully designed to optimize performance and efficiency for specific (predictable) situations. However, such an approach does not work well in recent complicated systems, because it takes long time or sometimes impossible to recover from a large-scale network failure, especially, unknown type of failures. Therefore, to cope with unpredictable changes of the system environment, e.g., system failures and change of system inputs, another design approach is required, in which each component in the system behaves to maintain a stable state and adapts for the environmental change flexibly and autonomously. By this approach, the system can maintain a stable state and offer a service of good quality in a highly dynamic environment, although the performance may not be optimized.

As the system becomes large-scale and more complicated, it becomes impractical to know in advance all events happened in the system and their factors, e.g., reasons and how to deal with them.

Fuzzy reasoning (Lee 1972) accumulates human knowledge as a knowledge database with *If-Then-Else* rules and performs recognition, control, and reasoning by the reasoning engine. While it is based on human experiences, it still requires advanced construction of rules, and thus, the above-mentioned problem is not solved.

Neural networks (Haykin 1989) perform pattern recognition by optimizing parameters in neurons. However, since they are based on machine learning, they cannot quickly adapt to a situation that has never been met and is hard to be predicted.

On the other hand, a genetic algorithm (Goldberg 1992) converts engineering data into the form of gene code and optimizes the system by imitating heredity processes that occur in organisms such as mutation, recombination, and optimal choice. Since it has both two aspects of random search and optimal choice, it is different from approaches that cannot get away from the local minimum such as the steepest descent method (Brooks et al. 1983). However, it basically assumes a well-formulated system, and thus, it cannot handle unknown changes occurred in the system.

Simulated annealing (Kirkpatrick et al. 1983) is an approach that examines multiple neighboring solutions of the current solution randomly and decides probabilistically which neighboring state to transit. It also has a mechanism to prevent from falling into the local minimum. However, since it searches for the optimal solution heuristically, it generally takes much time to converge and cannot cope with frequent changes of the environment.

As mentioned above, conventional approaches cannot fully cope with unknown changes in the system flexibly and quickly.

On the other hand, attractor selection has advantages that it can cope with unknown changes, and its calculation time is much shorter than conventional heuristic approaches. Since the contents of broadcast data are continuously changing in our assumed system environment, attractor selection is suitable for the problem of determining the order of filters which we address in this chapter.

8.4 Proposed Methods

In this section, we propose four methods that can adapt to the change of broadcast contents and reduce the filtering cost by using attractor selection to control the parameters in deciding the order of filters.

8.4.1 Attractor Selection (AS) Method

In the AS method, a client determines *filter selection priority S* by using attractor selection, where *S* consists of a list of filters and defines the order of applying the fil-

ters. Here, let us denote n as the number of applying filters and $S_{i,j}$ as the filter selection priority of applying filter F_i $(i = 1, 2, \ldots, n)$ at jth position $(j = 1, 2, \ldots, n)$ in S.

In the following, we describe the AS method in detail.

Calculation of the Filtering Cost We define the ratio of data items which are discarded by applying filter F_i as *decrease ratio* D_i. D_i is calculated by the following equation using the number of data items, d_i, discarded by F_i, and the number of data items, a_i, that are applied F_i:

$$D_i = \frac{d_i}{a_i}. \tag{8.6}$$

When a client uses the AS method in a real environment, it actually applies the filters to the broadcast data items and uses the elapsed time to perform filtering as the filtering cost. However, in our simulation evaluation, a client cannot apply filters actually since we use pseudo-data. Thus, we generalize the filtering cost and define the processing cost of each filter F_j as c_j. c_j represents the processing time per data item when a client applies F_j.

The total cost C for applying n kinds of filters in a certain order to N data items is calculated by the following equation:

$$C = \sum_{j=1}^{n} \left(c_j N \prod_{k=1}^{j-1} D_k \right). \tag{8.7}$$

Calculation of the Activity In the AS method, the activity α is defined by using C, since the system performance is considered better when the filtering cost is lower. Here, the minimum value of C among the last x results of filtering is denoted by $C_{\min}$. Then, the activity α is calculated by the following equation, where the activity becomes higher when the filtering cost approaches the minimum cost:

$$\frac{d\alpha}{dt} = \delta \left(\left(\frac{C_{\min}}{C} \right)^{\lambda} - \alpha \right). \tag{8.8}$$

Here, δ and λ are scale factors to control the adaptation rate and the value of activity. Note that α ranges $0 \leq \alpha \leq 1$.

Calculation of the Selection Priority The selection priority $S_{i,j}$ is defined by the following equation, which comes from the approaches in Leibnitz et al. (2005):

$$\frac{d}{dt} S_{i,j} = \frac{syn(\alpha)}{1 + S_{\max,j}^2 - S_{i,j}^2} - deg(\alpha) S_{i,j} + \eta_{i,j}, \tag{8.9}$$

$$syn(\alpha) = \alpha \left[\beta \alpha^{\gamma} + \phi^* \right], \tag{8.10}$$

$$deg(\alpha) = \alpha, \tag{8.11}$$

$$\phi(\alpha) = \frac{syn(\alpha)}{deg(\alpha)}, \tag{8.12}$$

$$\phi^* = \frac{1}{\sqrt{2}}. \tag{8.13}$$

Here, $\eta_{i,j}$ is a random number, and β and γ are scale factors (constants). $S_{i,j}$ ranges $0 \leq S_{i,j}$. Note that for $j \geq 2$, $S_{i,j}$ is set as 0 if F_i is already selected, since it is meaningless to apply the same filter more than once.

When the filtering cost is low and the activity is high, the selection priority rarely changes since the first term on the right side in (8.9) becomes dominant. However, when the filtering cost becomes high and the activity becomes low, the third term, i.e., noise, becomes dominant, and the system tries to transit another stable state. That is, the system adapts to the change of the broadcast contents.

Flow Chart Figure 8.5 shows the flow chart of the procedure performed by a client every time when N data items are stored in its receiving buffer in the AS method. We define one cycle of all the steps in this figure as a unit of filtering.

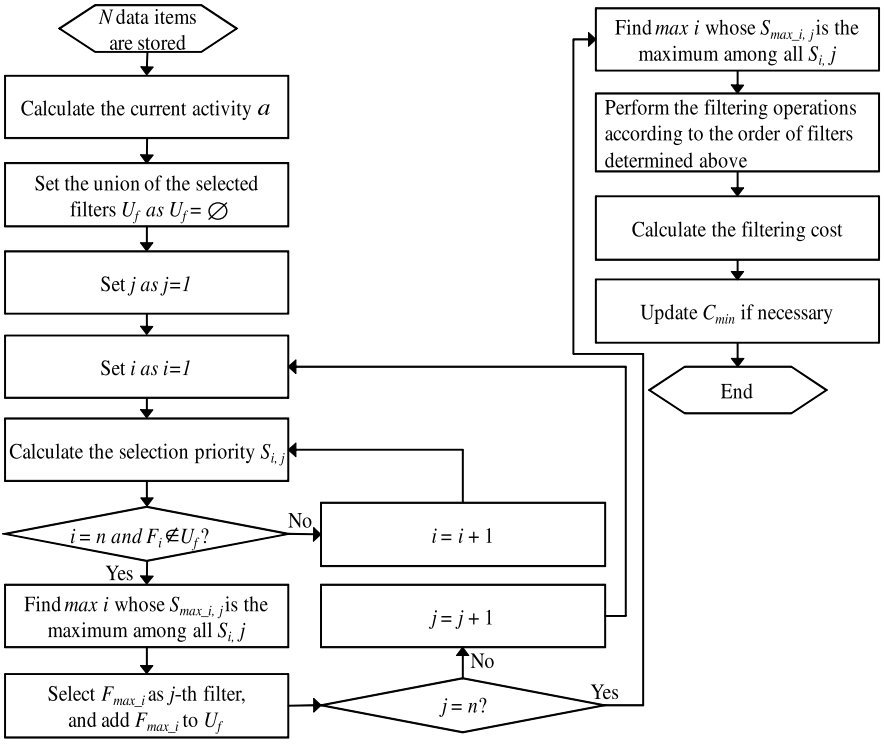


Fig. 8.5 Flow chart of the AS method

Problems of the AS Method Generally, a user's preference changes as time passes. When a user's preference changes, the condition of filtering changes (e.g., "I want to get the information about news, though I am receiving the information about sports"), and the number of filters also might change (e.g., "I want to get the information about baseball, though I am receiving the information about sports").

The AS method can deal with the changes in keywords for filtering and the parameters for the thresholds in filters in the same way as the case that broadcast contents change.

Moreover, the AS method can deal with the changes in the number of filters by adding or reducing the selection priority corresponding to these filters. However, in this case, the upper limit and the lower limit of the filtering cost change largely. Since (8.8) and (8.9) do not assume the changes of the upper and lower limits of the filtering cost, the AS method does not work well, i.e., the filtering cost becomes large. Similarly, the AS method does not work well when the processing costs of filters change. This is because the upper and lower limits of the filtering cost also change.

8.4.2 Extended Methods

In this subsection, we extend the AS method and propose three filtering order adaptation methods that consider the changes in the processing cost and the number of filters.

8.4.2.1 AS-M Method

In the AS method, the filtering cost increases when the processing cost c_i or the number of filters is not constant. This is because the filtering cost changes largely according to the changes in the order of the filters and the characteristics of the broadcast contents, which affect the ratio of $C_{\min}$ to C in (8.8). To solve this problem, we extend the AS method by adjusting λ in (8.8) to keep the value of $(C_{\min}/C)^{\lambda}$ as constant.

Specifically, the AS-M (AS using the Maximum filtering cost) method keeps $(C_{\min}/C)^{\lambda}$ as constant by changing λ according to the following equation:

$$\left(\frac{C_{\min}}{C_{\max}}\right)^{\lambda} = 0.02. \tag{8.14}$$

Here, $C_{\max}$ denotes the maximum value of C in the past. Note that 0.02 is found to be an appropriate value from our preliminary experiments.

In this way, the AS-M method controls the impact of the change of the upper and lower limits of the filtering cost in (8.8). Specifically, when the lower limit of the filtering cost becomes smaller and $C_{\min}/C$ in (8.8) becomes smaller, the impact of the change in the lower limit can be eliminated by changing λ smaller according

to (8.14). On the other hand, when the lower limit of the filtering cost becomes larger and $C_{\min}/C$ in (8.8) becomes larger, the impact of the change in the lower limit can be eliminated by changing λ larger.

As a result, the AS-M method can calculate the activity adequately and deal with the changes in the processing costs of filters and the number of the filters related to the change of the user's preference.

8.4.2.2 AS-P Method

The AS-M method can deal with the change of the upper and lower limits of the filtering cost by changing λ in (8.8). However, from (8.8) and (8.9) it is shown that the change of the upper and lower limits can be also handled by changing not λ but $\eta_{i,j}$.

Therefore, the AS-P (AS with Perturbation) method adjusts the balance of the random term in (8.9). This is based on the approach called *attractor perturbation* (Leibnitz et al. 2009), which is an operation method of the random term.

The random term in the AS-P method, $\eta(t)$, is calculated by the following equation:

$$\frac{d\eta(t)}{dt} = \tau\big[\eta\big(1-\alpha(t)\big) - \eta(t)\big]. \tag{8.15}$$

Here, τ denotes the parameter to define the balance of the random term.

The AS-P method controls the random term instead of using the constant noise term $\eta_{i,j}$ in (8.9). According to (8.15), the impact of the random term becomes small when the activity is high, and it becomes large when the activity is low.

As a result, the AS-P method can make the activity more stable when the activity is high. Also, it can transit to another stable state more easily when the activity is low, compared with the AS method.

8.4.2.3 AS-MP Method

Since the AS-M and the AS-P methods can work independently with each other, we suppose that we can further reduce the filtering cost by applying both methods together. Therefore, the AS-MP method combines the AS-M and AS-P methods, where λ and the random term are respectively adjusted according to the AS-M and AS-P methods.

8.5 Evaluation

This section evaluates the proposed methods using simulation studies. The evaluation criterion is the *average filtering cost*, which is the average of the filtering costs for all filtering processes performed during the simulation time.

8.5.1 Simulation Environment

Table 8.1 shows the parameters used in the simulations. In the simulations, the broadcast data and the filtering model are assumed as an information service for mobile clients as described in Sect. 8.2. Moreover, we did not use real broadcast data but use pseudo data to represent various situations by changing parameters. To apply multiple different filters, we attach the same number of attributes as filters to the pseudo-data and the attribute values (e.g., keywords) are used for filtering.

For simplicity, all filters perform selection operations, i.e., only data items that contain the attribute values specified by the client are stored, and other items are discarded. Thus, the filtering results do not depend on the order of applying filters (Sawai et al. 2004). The results of our simulations correspond to the results in other settings where filters whose applying order does not affect the filtering results are assumed. We can also easily consider cases where filters whose applying order does affect the filtering result by introducing a mechanism to take dependencies of the applying order into account when determining the order of filters.

The distribution of attribute values attached to the pseudo-data is determined according to the Zipf distribution. Here, many conventional studies on broadcast information systems also assume the Zipf distribution for distribution of data values (Acharya et al. 1995; Aksoy and Franklin 1998). The Zipf distribution is shown in the following equation:

$$f(r) = \frac{\frac{1}{r}}{\sum_{m=1}^{N_a} \frac{1}{m}}. \tag{8.16}$$

Table 8.1 Parameters

Parameter	Value
Number of times of filtering	50000
Number of filters	5
Number of tags	5
Number of keywords	5
Size of the receiving buffer	5000
Calculation cycle in the optimal method	10000
Calculation cycle in the genetic algorithm	7000
β	0.4
γ	5.0
δ	3.0
η	−0.1–0.1
λ	10
Number of steps in the Runge–Kutta method	10
Bandwidth of the broadcast channel [Mbps]	10
Size of a data item [kByte]	1

Here, N_a denotes the number of attribute values, and r represents the rank when the attribute values are ordered sequentially by the number of data items having that attribute value. $f(r)$ represents the probability that the attribute value of rank r appears. When rank r is low, $f(r)$ becomes high, and the number of data items having that attribute value becomes high. In this chapter, we assume that the attribute values consist of not only the keywords and the categories but also the numerical values such as cosine correlation. When the attribute values are numerical values, it is assumed that the ratio of the number of data items that belong to each interval among the uniformly divided intervals within the total range of the attribute values follows the Zipf distribution.

In the simulations, we change r for each attribute value randomly as time passes. This represents the change of the broadcast contents. We call the cycle of changing r for each attribute value as the *attribute rank changing cycle* and the timing of changing r as the *attribute rank changing timing*. We change the attribute rank changing cycle from 600 to 1400. Moreover, we basically set the processing cost c_i of filter F_i $(i = 1, 2, \ldots, 5)$ as $c_i = 0.6$ $(i = 1, 2, \ldots, 5)$ [ms/data].

It is assumed that the user specifies an attribute value that represents the user's preference to each attribute. The data items whose attribute values match the user's preference are stored, and others are discarded. One filter selects data items whose attribute value matches the user's one attribute. We also assume that the number of attributes attached to each data item equals the number of filters.

In the simulations, we set β, γ, δ, η, and λ as values determined by some preliminary experiments. In the AS-P method, η is set to $-0.5 < \eta < 0.5$. Furthermore, we set $C_{\min}$ as the minimum cost of the past 50 times of filtering.

8.5.2 Comparison Methods

In our simulations, we compared our proposed methods with the following four methods.

Minimum Cost Method: In the minimum cost method, a client calculates the filtering cost for each of all possible $n!$ kinds of orders of filters for every filtering process and adopts the order that gives the minimum cost among them. Note that this method is unrealistic because the computation load is too high to apply it in a real environment. Therefore, we show the performance of this method as a lower bound.

Cyclic Adaptation Method: At a certain calculation cycle, a client once performs filtering using each of all possible $n!$ kinds of orders of filters at the cycle and adopts the order that gives the minimum cost among them. Then, the client adopts the same order of filters until the next calculation cycle. In this method, the filtering cost at the calculation cycle becomes equal to that of the minimum cost method but cannot adapt to the change of broadcast contents until the next cycle.

Note that the load at the calculation cycle is high, since the client has to calculate the costs of all $n!$ kinds of orders. We define the value obtained by dividing the

filtering cost to perform this method once by the calculation cycle as the *optimal order calculation cost*. Moreover, we call the sum of the average filtering cost and the optimal order calculation cost as the *total cost*.

Genetic Algorithm: In the genetic algorithm, similar to the cyclic adaptation method, a client periodically searches an appropriate order of filters according to the genetic algorithm so that the filtering cost becomes less. Then, the client adopts the same order of filters until the next calculation cycle. Specifically, a client selects the order of filters that provides the minimum cost after performing filtering using each of several kinds (much less than $n!$) of orders among $n!$ kinds of orders, while all $n!$ kinds of orders are examined in the cyclic adaptation method. The calculation cost of the genetic algorithm is lower than that of the cyclic adaptation method. However, it cannot always find the order with the minimum cost.

We set the crossover rate as 0.8, the mutation rate as 0.03, the number of children as 10, and the maximum number of generation as 6. Moreover, we use the combination of elitist selection and roulette selection as the selection method, uniform the crossover as crossover method, and inversion mutation as the mutation method.

Random Method: In the random method, a client decides the order of filters at random for every filtering process.

8.5.3 Evaluation Criteria

We use the following three costs as criteria for the evaluation. The unit of all the criteria is millisecond.

Filtering Cost: The time to perform the filtering operations. In our simulation evaluation, a client cannot apply filters actually since we use pseudo-data. Thus, we generalize the filtering cost as (8.7) based on some preliminary experiments. We use this criterion only in Sect. 8.5.4.2.

Average Filtering Cost: The average of filtering costs of all filtering processes performed during the simulation experiments. In other words, it represents the average cost per filtering process.

Total Filtering Cost: The total sum of the average filtering cost and the average calculation cost. The average calculation cost is defined as the average of costs for calculating the order of filters at the calculation cycle in each method. This is based on the actual calculation times in some preliminary experiments. Here, in the AS method and the extended methods, the calculation cost (i.e., the cost for calculating variables such as the activity and the selection priority) is ignored since it is much smaller than the average filtering cost.

8.5.4 Simulation Results

8.5.4.1 Impact of Calculation Cycle in the Cyclic Adaptation Method

Figure 8.6 shows the average filtering cost, the optimal order calculation cost, and the total cost of the cyclic adaptation method when the calculation cycle changes from 1000 to 12000.

The result shows that the average filtering cost when the calculation cycle is 1000 is the lowest. This is because the calculation cycle is short and, thus, the server can adapt to the change of broadcast contents rapidly. The average filtering cost basically becomes higher as the calculation cycle gets longer.

On the other hand, the optimal order calculation cost becomes lower as the calculation cycle gets higher. This is because the longer the calculation cycle is, the less the number of times of calculating the costs is.

The total filtering cost, the sum of the average filtering cost and the optimal order calculation cost, is the lowest when the calculation cycle is 10000. This shows that the average filtering cost and the optimal order calculation cost have a trade-off relation, and the system performance is balanced when the calculation cycle is 10000 in this simulation environment. Thus, we chose 10000 as the calculation cycle in the cyclic adaptation method in the following experiments.

8.5.4.2 Comparison among Methods

Figure 8.7 shows the total filtering cost of each method. From this result, the total filtering costs of the four proposed methods are lower than that of the cyclic adaptation method, the genetic algorithm, and the random method. The average filtering cost of the cyclic adaptation method is slightly lower than the random method. However,

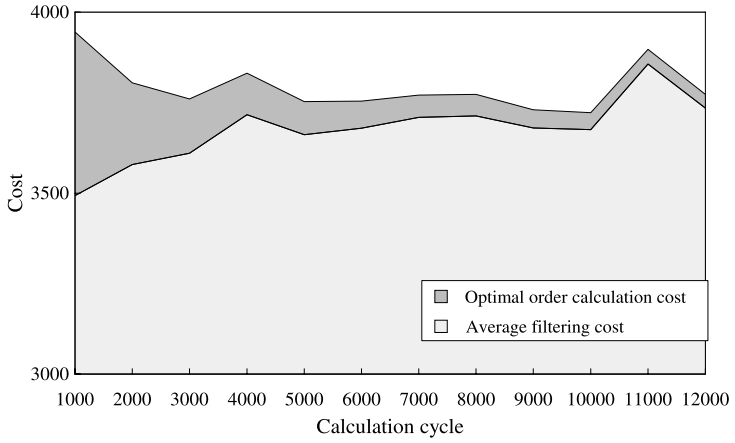


Fig. 8.6 Impact of calculation cycle in the cyclic adaptation method

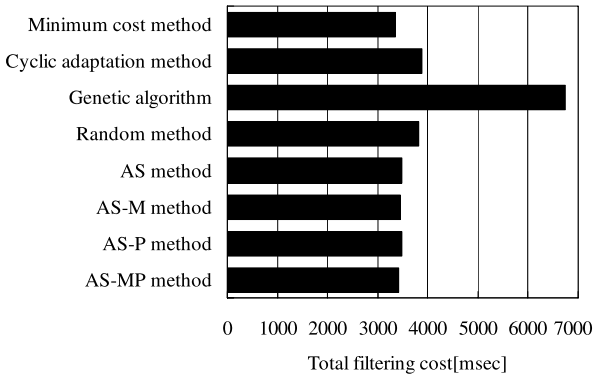


Fig. 8.7 Comparison between methods

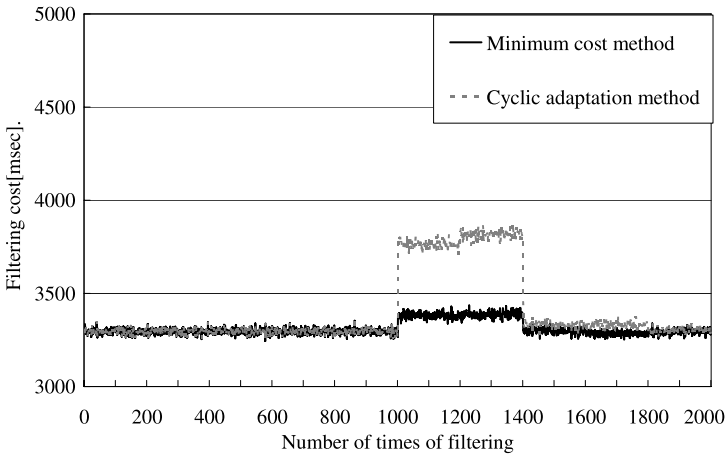


Fig. 8.8 Transition of the filtering cost of the minimum cost method and the cyclic adaptation method

the cyclic adaptation method requires an extra cost to calculate the order of filters, i.e., the optimal order calculation cost, and thus, the total filtering cost is higher than the cost of the random method.

The total filtering cost of the genetic algorithm is much higher than that of the cyclic adaptation method. In the genetic algorithm, the optimal order calculation cost at the calculation cycle is lower than that in the cyclic adaptation method. However, the average filtering cost of filters whose order is determined in the calculation cycle is often far from the minimum. Thus, the total filtering cost of the genetic algorithm becomes higher than that of the cyclic adaptation method.

Figure 8.8 shows the transition of the filtering costs of the minimum cost method and the cyclic adaptation method, and Fig. 8.9 shows the transition of the filtering costs of the AS method and the genetic algorithm. Figures 8.10, 8.11, and 8.12

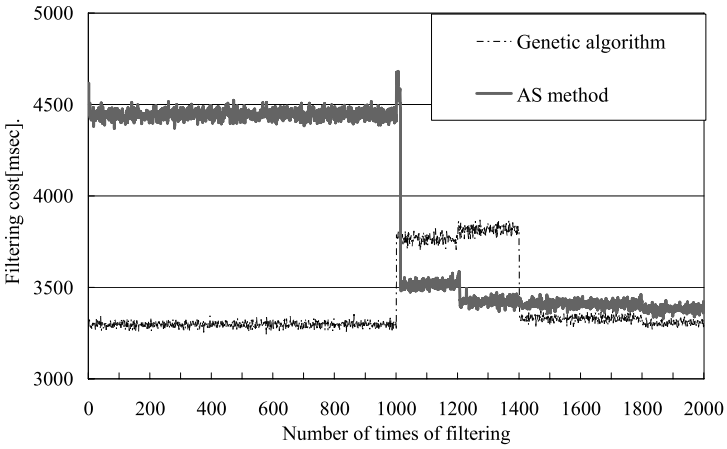


Fig. 8.9 Transition of the filtering cost of the genetic algorithm and the AS method

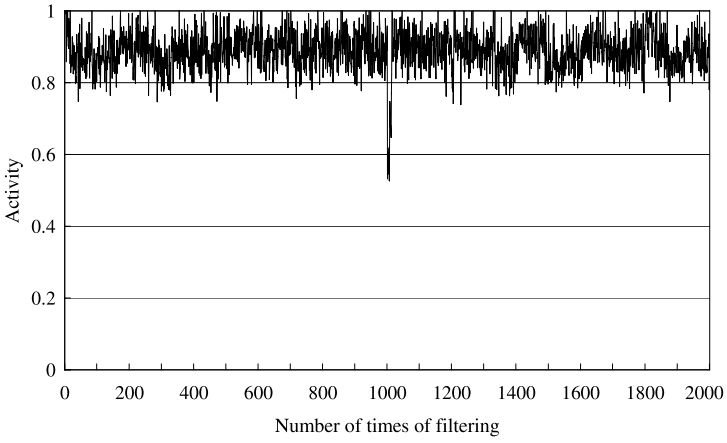


Fig. 8.10 Transition of the activity in the AS method

show the transitions of the activity and the selection priority $S_{i,1}$ $(i = 1, 2, \ldots, 5)$ in the AS method. Due to the limitation of space, we only show the results from the simulation starting time to the time until 2000 times of filtering processes are performed.

From Figs. 8.8 and 8.9, it is shown that the filtering cost of every method except for the minimum cost method changes largely after the time when 1000th filtering process is performed at which broadcast contents change. However, in the AS method, the filtering cost becomes low soon, which shows that our method can adapt to the change of broadcast contents.

Figure 8.10 shows that the activity becomes very low when the broadcast contents change and the filtering cost becomes high. Moreover, Figs. 8.11 and 8.12 show

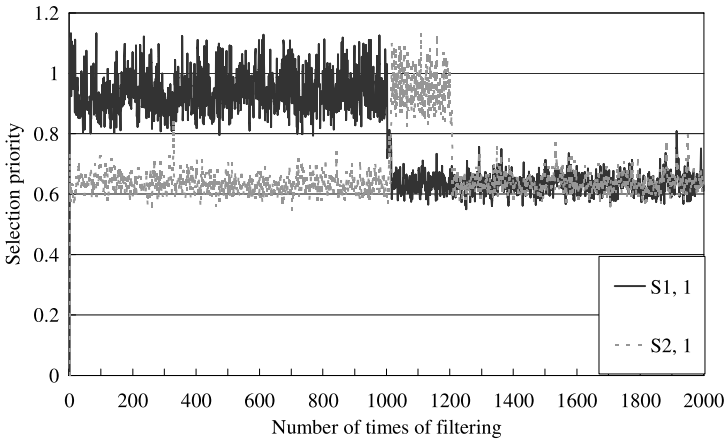


Fig. 8.11 Transition of the selection priority in the AS method ($S_{1,1}$, $S_{2,1}$)

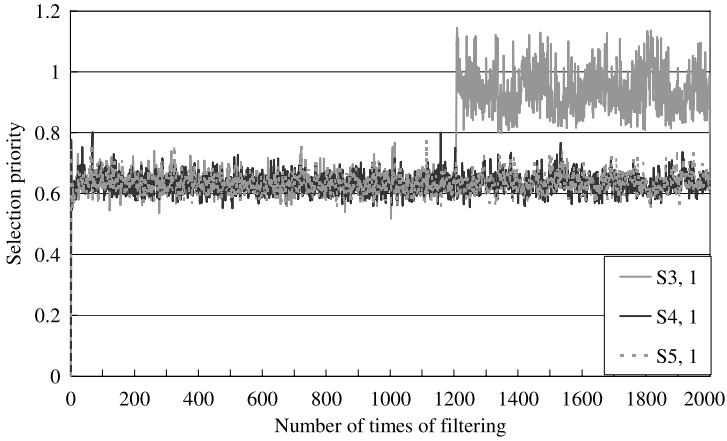


Fig. 8.12 Transition of the selection priority in the AS method ($S_{3,1}$, $S_{4,1}$, $S_{5,1}$)

that when the activity becomes low, the random term in (8.9) influences largely, and thus, the selection priority goes up and down greatly. After a short time, the system transits into a stable state, and the selection priority of a specific filter rises.

In summary, in the AS method, the client changes the order of filters adaptively by using attractor selection to control the selection priority when the broadcast contents change and the filtering cost becomes high. It confirms us the effectiveness of using attractor selection to adapt to the change of the broadcast contents. Here, in the AS method, the activity sometimes becomes low, and the order of filters is changed even when broadcast contents do not change. This is due to the influence of the random term in (8.9).

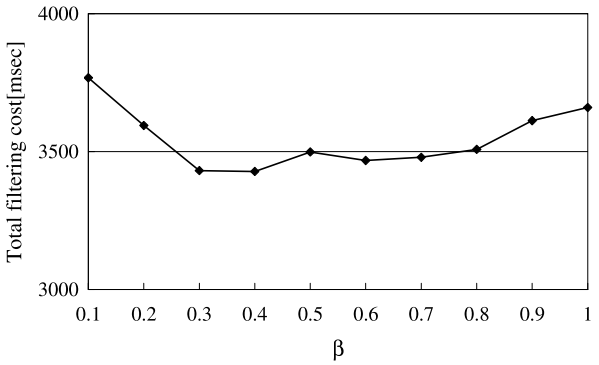


Fig. 8.13 Impact of β

In this simulation setting (as expected), the three extended methods (AS-M, AS-P, and AS-MP) show almost the same performance as the AS method.

8.5.4.3 Impact of β on the AS Method

Figure 8.13 shows the total filtering cost of the AS method when β changes from 0.1 to 1.0. From this result, the total filtering cost is low when $\beta = 0.3$ to 0.8. Here, β is a constant that coordinates the influence of the random term in (8.9). When β is very high, the random term influences little, so that the selection priority and the order of filters do not change even when the filtering cost is high and the activity is low. On the other hand, when β is very low, the random term influences largely, and the AS method acts similarly to the random method.

8.5.4.4 Impact of γ on the AS Method

Figure 8.14 shows the total filtering cost of the AS method when γ changes from 1 to 10. From this result, the total filtering cost is low when $\gamma = 4$ to 7. Here, γ is a constant that coordinates the influence of the activity in (8.9). The influence of the activity becomes small when γ is high. However, in our simulations, the total filtering cost is not much influenced by γ.

8.5.4.5 Impact of δ on the AS Method

Figure 8.15 shows the total filtering cost of the AS method when δ changes from 1 to 7.

The result confirms that the total filtering cost hardly changes even if δ changes, i.e., the impact of δ is very small.

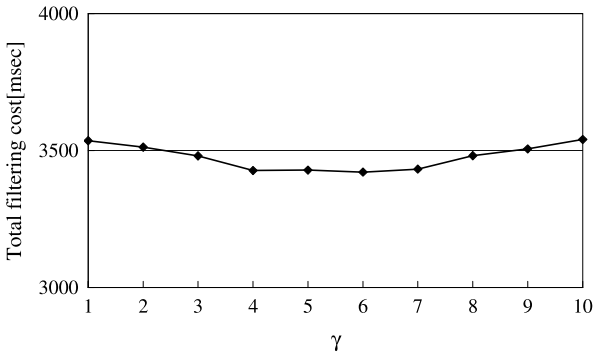


Fig. 8.14 Impact of γ

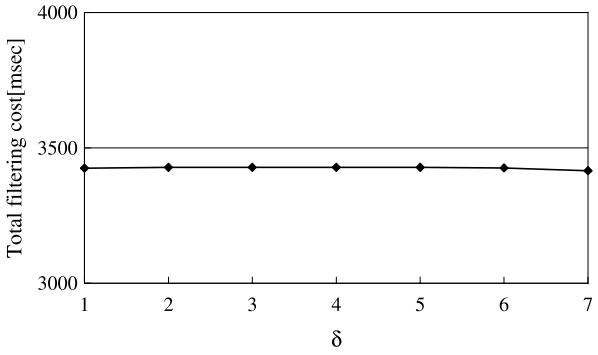


Fig. 8.15 Impact of δ

8.5.4.6 Impact of x on the AS Method

Figure 8.16 shows the total filtering cost of the AS method when x changes from 10 to 100. Here, x is the window size for calculating $C_{\min}$, i.e., the AS method determines $C_{\min}$ as the minimum filtering cost among the last x filtering processes.

From this result, the total filtering cost is low when $x = 30$ to 100. If x is very small, $C_{\min}$ is updated frequently, and the activity tends to be unstable according to (8.8). On the other hand, if x is large, $C_{\min}$ is rarely updated even when the broadcast contents change, and thus, the activity also tends to be unstable.

8.5.4.7 Impact of c_i

Figure 8.17 shows the total filtering cost of each method when changing the processing cost c_i randomly between 0.1 to 2.0 [ms/data] at every 1000th filtering. This figure shows that the AS-M and the AS-P methods reduce the total filtering cost compared with the AS method, the minimum cost method, the cyclic adaptation method, the genetic algorithm, and the random method. Moreover, the AS-MP

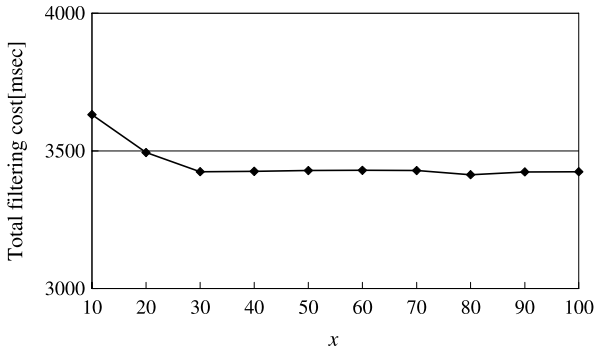


Fig. 8.16 Impact of x

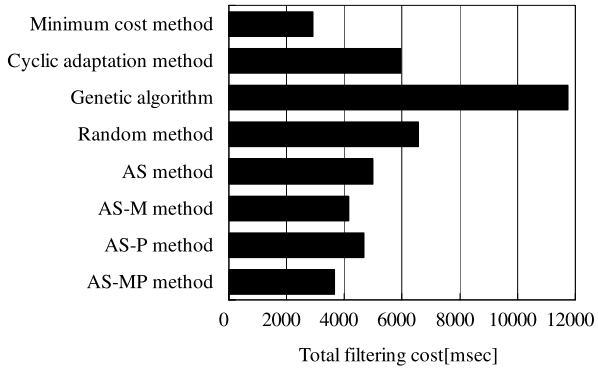


Fig. 8.17 Impact of c_i

method which combines the AS-M and the AS-P methods gives the lowest total filtering cost.

The AS method does not take into account the change of the upper and lower limits, and thus, the order of filters sometimes changes even if the filtering cost is almost same as the lower limit. This is because, according to (8.8), the influence on the activity becomes large when λ is not appropriately chosen.

The AS-M method can calculate the activity appropriately by controlling λ based on the max value of the past filtering costs, even if the processing cost changes. Thus, the AS-M method can adapt to the change in the processing cost.

The AS-P method reduces the total filtering cost, since it can adapt to the change of the filtering cost flexibly by controlling the random term in (8.9). However, the effect of the AS-P method is lower than the AS-M method, since the AS-P method only adjusts the random term in (8.9), while the AS-M method feeds back the change of the upper and lower limits to the activity.

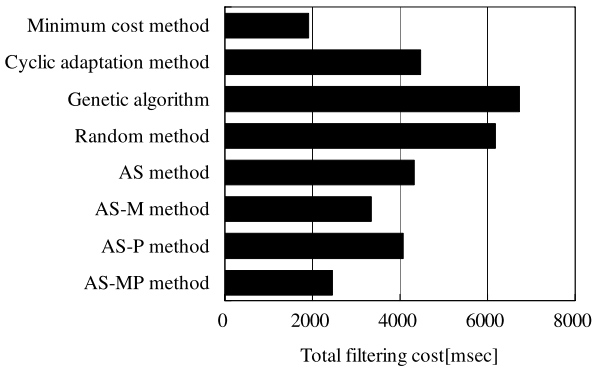


Fig. 8.18 Impact of the number of filters

8.5.4.8 Impact of the Number of Filters

Figure 8.18 shows the total filtering cost of each method when changing the number of filters randomly between 2 to 5 at every 1000th filtering. In this simulation, we set the processing cost c_i as $c_i = [0.2, 0.6, 1.0, 1.4, 1.8]$. Moreover, the number of attributes is set as the number of filters.

From this result, it is also shown that the AS-M and AS-P methods reduce the total filtering cost compared with other methods. Moreover, the AS-MP method gives the lowest total filtering cost. This result confirms us that the extended methods can reduce the total filtering cost by controlling λ and the random term dynamically even when the number of filters changes.

8.6 Conclusions

In this chapter, we proposed a novel method called the AS method that uses attractor selection to control parameters to determine the order of applying filters. With the proposed method, the client adaptively changes the order of filters following the change of broadcast contents to reduce the filtering load. Moreover, we extend the proposed method and propose three methods that take into account the change of the user's preference. The simulation results confirmed us that the proposed methods reduce the filtering cost compared with other methods except for the minimum cost method (lower bound). Furthermore, the extended methods further reduce the filtering cost compared with the AS method.

As part of our future work, we plan to examine the influence of the change of broadcast contents and user's preference on the performance of our methods in more detail.

Acknowledgements This research was partially supported by "Global COE (Centers of Excellence) Program" and Special Coordination Funds for Promoting Science and Technology "Formation of Innovation Center for Fusion of Advanced Technologies: Yuragi Project" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Acharya, S., Alonso, R., Franklin, M., and Zdonik, S.: Broadcast disks: Data management for asymmetric communication environments, in *Proceedings of ACM SIGMOD 1995*, pp. 199–210, May 1995.

Aksoy, D. and Franklin, M.: Scheduling for large-scale on-demand data broadcasting, in *Proceedings of IEEE The Conference on Computer Communications (INFOCOM 1998)*, pp. 651–659, Mar. 1998.

Belkin, N. J. and Croft, W. B.: Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, Vol. 35, No. 12, pp. 29–38, 1992.

Bell, T. A. H. and Moffat, A.: The design of a high performance information filtering system, in *Proceedings of SIGIR 1996*, pp. 12–20, Aug. 1996.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M.: CHARMM: A program for macromolecular energy, minimization, and dynamics calculations, *Journal of Computational Chemistry*, Vol. 4, pp. 187–217, 1983.

Goldberg, D. E.: Genetic algorithms in search, optimization and machine learning, *Communications of the ACM*, Vol. 35, No. 12, pp. 29–38, 1992.

Haykin, S.: Neural Networks: A Comprehensive Foundation, Addison-Wesley, Reading, 1989.

Kashiwagi, A., Urabe, I., Kaneko, K., and Yomo, T.: Adaptive response of a gene network to environmental changes by fitness-induced attractor selection, *PLoS ONE*, Vol. 1, No. 1, e49, Dec. 2006.

Kitajima, S., Hara, T., Terada, T., and Nishio, S.: Filtering order adaptation based on attractor selection for data broadcasting system, in *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009)*, pp. 319–326, Mar. 2009.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P.: Optimization by simulated annealing, *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.

Lee, R. C. T.: Fuzzy logic and the resolution principle, *Journal of the ACM*, Vol. 19, No. 1, pp. 109–119, 1972.

Leibnitz, K., Wakamiya, N., and Murata, M.: Biologically inspired adaptive multi-path routing in overlay networks, in *Proceedings of IFIP/IEEE International Workshop on Self-Managed Systems & Services (SelfMan 2005)*, (CD-ROM), May 2005.

Leibnitz, K., Furusawa, C., Murata, M.: On attractor perturbation through system-inherent fluctuations and its response, in *Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA 2009)*, (CD-ROM), Oct. 2009.

Salton, G. and McGill, M. J.: Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

Sawai, R., Tsukamoto, M., Terada, T., and Nishio, S.: Composition order of filtering functions for information filtering, in *Proceedings of International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2004)*, pp. 166–171, Jan. 2004.

Chapter 9
StreamAPAS: Query Language and Data Model

Marcin Gorawski and Aleksander Chrószcz

Summary The system StreamAPAS and its declarative query language allows users to define temporal data analysis. This chapter addresses the problem of lack of the continuous language standard. The proposed language syntax indicates how hierarchical data structures simplify working with spatial data and groups of tuple attributes. The query language is also based on object-oriented programming concepts as a result of which continuous processing applications are easier to develop and maintain. In addition, we discuss the problem of a query logic representation. In contrast to relations stored in DBMS, data streams are temporal so that DSMS should be aware of their dynamic characteristics. Streams characteristics can be described using variables such as tuple rates and invariables like monotonicity. In StreamAPAS, a query is represented as a directed acyclic graph (DAG) whose operators define tuple data transmission model and have information of result stream monotonicity associated with them. Even though this representation is still static, this approach enables us to detect optimization points which are crucial from a stream processing viewpoint.

9.1 Introduction

In the chapter we present the stream processing architecture of StreamAPAS and the prototype query language. There are a lot of research projects which develop a declarative query language for data stream processing (e.g., Babcock et al. 2002; Ali et al. 2005; Yan-Nei et al. 2004). In these researches authors create query languages which syntax bases mainly on the SQL. As a result, users are able to define query upon streams and relations in a convenient way. We should remember

M. Gorawski (✉) · A. Chrószcz
Institute of Computer Science, Silesian University of Technology, Akademicka 16,
44-100 Gliwice, Poland
e-mail: Marcin.Gorawski@polsl.pl

A. Chrószcz
e-mail: Aleksander.Chroszcz@polsl.pl

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_9,

that the SQL is mainly intended for expressing simple queries which are processed over a single database. The data stream queries are different in two aspects. Data Stream Management System (DSMS) usually connects to remote sources and sinks. Besides, the stream processing applications usually need the implementation of custom functions. In our research, we concentrate on the problem of finding the abstract elements that should be introduced to the query language so that the language functionality can be easily adapted to the application requirements.

In defining the semantics and concrete language the following goals were considered:

- Constructing a query language which allows users to define custom functions, import them into the stream processing platform, and use them as native language functions,
- Using a hierarchical data structure, which better suits spatial and analytical data representation,
- Defining a stream processor based on temporal logical operator algebra (Krämer and Seeger 2005), which offers efficient stream-to-stream physical operators.

In contrast to STREAM (Babcock et al. 2002) and ATLAS (Yan-Nei et al. 2004) which are mainly based on the SQL syntax, we added to the query language elements of Object-Oriented (OO) languages. We adopt from OO languages calling class functions, calling object functions, and loading user-defined libraries into the compiler. Thanks to this, the user is able to define new data sources, custom operators, and data sinks by calling library functions from the level of the query language.

The remaining part of this chapter is organized as follows: Sect. 9.2 describes the data stream processing implemented in SreamAPAS; Sect. 9.3 introduces its query language and implementation aspects; next, in Sect. 9.4 we compare our language with CQL language; Sect. 9.5 shows the aims of our further work; and finally Sect. 9.6 presents a summary of our results.

9.2 Data Stream Processing

We use the directed acyclic graph (DAG) to describe the data stream query. DAG nodes represent data stream operators, and edges define stream connections between the operators. We distinguish two levels of a query definition. On the logical level, DAG nodes represent operators of the logic operator algebra (Krämer and Seeger 2005). On the physical level, DAG node describes which algorithm is used to compute a given logical operator, and DAG edge defines how the data communication is implemented. There are a number of data stream processor architectures (Tucker 2005; Krämer and Seeger 2005; Motwani et al. 2003; Abadi et al. 2003). In contrast to them, we develop a stream processor architecture which processes temporal tuples (Krämer and Seeger 2005; Krämer 2007) and positive/negative tuples (Ghanem et al. 2005). Let T be a discrete time domain. Let $I := \{[t_s, t_e) \mid t_s, t_e \in T \wedge t_s \leq t_e\}$ be the set of time intervals.

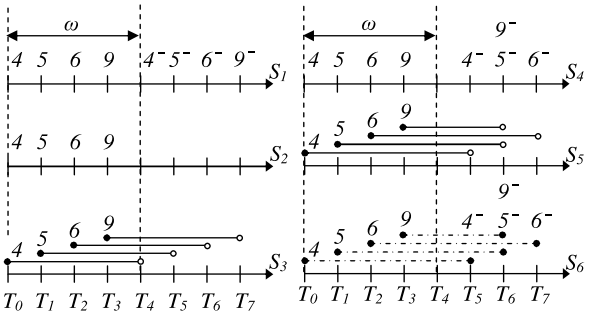


Fig. 9.1 Different stream definition approaches

Definition 9.1 (Stream) A pair $S = (M, \leq_{t_s,t_e})$ is a stream if: M is an infinite sequence of tuples $(type, e, [t_s, t_e])$, where: *type*—tuple type, *e*—attribute tree data, $[t_s, t_e) \in T$. $\leq_{t_s,,t_e}$ is the lexicographical order relation over M (primarily sorting by t_s and secondarily by t_e).

We can calculate the set of valid tuples for a given data stream and a specified point in time t. This set of tuples can be represented in the relational data base as table records which are available in time t. Streams S_1, S_2, and S_3 in Fig. 9.1 illustrate ways of controlling the lifetimes of those records. Stream S_1 uses two tuples so as to define the lifetime of a record. A positive tuple signals the beginning of a record existence, and a negative one points the end. When those lifetimes are constant and equal, we can only transmit the positive tuples as it is shown in stream S_2. Knowing the application time and the lifetime period, we can define a *time window* (Ghanem et al. 2005) which translates the input stream S_2 into the set of valid tuples. When we knew the lifetimes of table records at their time of creation, we can use temporal tuples (Krämer and Seeger 2005) which contain *start* and *end* timestamps as it is shown in S_3. The main advantage of temporal tuples is that they reduce the amount of transmitted data doubly. We cannot achieve this reduction in the model with positive and negative tuples, because their tuples contain only timestamp *start*. The stream S_4 cannot be reduced to positive tuples like S_1 to S_2 because the lifetimes of S_4 tuples are not functions of attribute *start*. However, we can apply the temporal tuples so as to remove negative tuples form the stream as it shows S_5. When we do not know the lifetimes of tuples at their time of creation, the temporal tuple model becomes useless. In some applications, it is acceptable to divide an entry time into the periods. Then, when a period elapses, temporal tuples which represent valid records or events are generated. Unfortunately, this solution generates a lot of additional data, and tuples are one time period delayed. In our research, we develop a system which joins the temporal stream model and the streams with positive and negative tuples.

We use the concepts of *positive temporal tuple* and *negative tuple*. When the lifetime of a table record or an event is known at the time of its generation, it is represented only by *positive temporal tuple*. The lifetimes of those tuples are represented in figures by solid lines. When we do not know the lifetime of the records or events at the time of their generation, we represent them by dashed lines. In such a

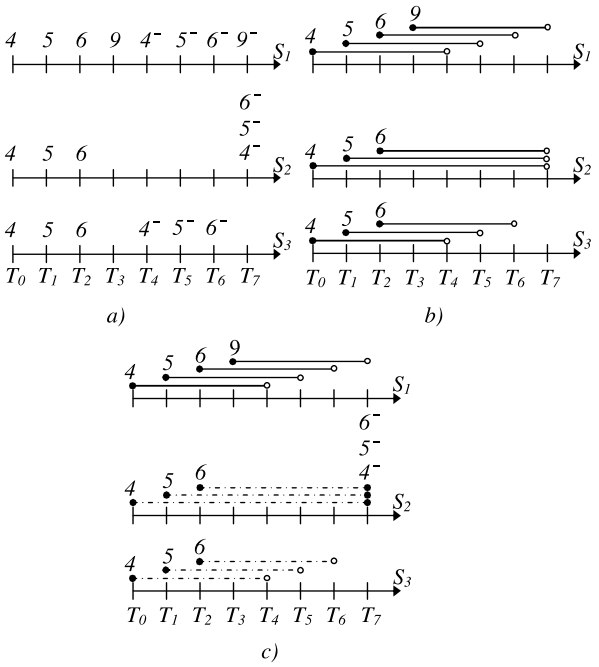


Fig. 9.2 Joining streams in different stream definition approaches

case, the timestamp *end* of a *positive temporal tuple* defines the upper boundary of the tuple lifetime. If we do not know when a tuple can expire, we assign the infinity value to timestamp *end*. When we know that a tuple will expire by a given time, we assign this value to timestamp *end*. The *negative tuple* expires the *positive temporal tuple* in identical way as it is described in model with positive and negative tuples (Ghanem et al. 2005). In comparison with *positive temporal tuple*, the *negative tuple* has zero lifetime period. Those tuples in figures are represented by points. The example of this model is stream S_6 which can substitute S_4.

In Fig. 9.2, we compare join operators for: (a) the model with positive and negative tuples; (b) the temporal model; and (c) the mixed model. Streams S_1 and S_2 are the input streams, and S_3 is the result of a join operator. Stream S_1 transmits only *positive temporal tuples*, and S_2 transmits both types of tuples in Fig. 9.2(c). Let us notice that the result streams S_3 in Fig. 9.2(b) and Fig. 9.2(c) transmit the same tuples, but they have a different interpretation. The result tuples in Fig. 9.2(b) define precisely their lifetimes, whereas the result tuples in Fig. 9.2(c) define the upper boundaries of their lifetimes. Let us note that we have negative tuples in input stream S_2 in example (c), however there is no negative result tuples. This situation happens because the negative tuples arrive at S_2 later than the upper lifetime boundary of result tuples.

In order that physical operators interpret correctly their input streams, each stream has defined monotonicity which is obtained from the operator connected to this stream input. In StreamAPAS, we borrow the stream monotonicity classification from (Golab 2006). Let Q be a query, and τ a point in time. Assume that at τ,

all tuples with lower or equal timestamps have been already processed. The multiset of input tuples at time τ is denoted $S(\tau)$, whereas all the tuples from time 0 to the current time are denoted by $S(0, \tau)$. Furthermore, let $P_S(\tau)$ be the result multiset produced at time τ, and let $E_S(\tau)$ be the multiset of expired tuples at time τ. The equation below defines the result set update function:

$$\forall \tau \quad Q(\tau + 1) = Q(\tau) \cup P_S(\tau + 1) - E_S(\tau + 1).$$

The types of stream monotonicity are defined indirectly. Using the above symbols, we define operators that generate stream of a given monotonicity:

1. The monotonic operator is an operator that produces result tuples which never expire. Formally the property is described by $\forall \tau \forall S E_S(\tau) = 0$.
2. The weakest nonmonotonic operator is an operator that produces result tuples whose lifetime is known and constant. Thanks to that, the order in which those tuples appear at the operator input correspond to the order of their expiration. Formally represented, it looks like $\forall \tau \forall S \exists c \in \mathrm{N} E_S(\tau) = P(\tau - c)$.
3. The weak nonmonotonic operator is an operator whose result tuples have different lifetimes but they are still known at the time of their generation. Let us note that the order of tuple insertion and the order of their expiration are different. This can be formalized as follows: $\forall \tau \forall S \forall S' S(0, \tau) = S'(0, \tau)$, it is true that $\forall t \in P_S(0, \tau) \exists et \in E_S(e) \wedge t \in E'_S(e)$.
4. The strict nonmonotonic operator is an operator whose expiration of tuples depends on the input tuples that will arrive in the future. The lifetimes of tuples are not known at the time of their generation. This can be formalized as follows: $\exists \tau \exists S \exists S' S(0, \tau) = S'(0, \tau)$ and $\exists e \exists t \in P_S(0, \tau)$ such that $t \in E_S(e) \wedge t \notin E'_S(e)$.

The monotonicity of type one says that the tuples of a given stream never expire. This means that the stream of this type transmits only *positive temporal tuples* with infinity assigned to *end* timestamp. The monotonicity of type two is illustrated by S_3 in Fig. 9.1. The stream S_5 in Fig. 9.1 is an example of stream with monotonicity of type three. The last type of monotonicity is illustrated by S_6 in Fig. 9.1. Those examples show that the type of stream monotonicity gives sufficient information to determine a tuple processing algorithm.

From the viewpoint of an operator monotonicity, there are two management types of result streams:

- The direct approach in which an operator calculates the lifetime of a result tuple directly at the time of the tuple generation. The tuple's lifetime is determined only via its timestamp *start* and *end*. Hence, the expired tuples can be determined using the application time without the need for negative tuples. Here we classify the operators defined over *time-based* windows.
- The negative tuple approach exists when an operator is assigned to the operator that generates a negative tuple or the operator is defined over a count type window (such as a *slide window*). As a consequence, an operator result stream has negative tuples. This management type has two disadvantages. The output streams of the operators have nearly twice as many tuples in comparison to result streams in

the direct approach. Moreover, operator *count type window* uses more memory resources.

Let us notice that the higher number of stream monotonicity, the more complicated architecture of the tuple collection which is connected to a given stream. Stream monotonicity of numbers: 1, 2, and 3 process no *negative tuples*. Those collections check only timestamp *end* so as to find expired tuples. If a stream is a weakest nonmonotonic one, the tuples expiration order is identical to the stream order. As a result, potentially expired tuples exist only at the beginning of tuple collection. In consequence, a simple list data structure is enough to implement this tuple collection. The stream monotonicity of type three has two potential implementations. The expired tuples can be identified by testing all the elements of a collection, or we can add an additional list in timestamp *end* order.

Suppose that there exists a query which consists of a few join operators. Because the join operators are commutative, we can change the order of their processing in a query plan. When we reduce the number of operators with high number of monotonicity type in a query plan production, then we also reduce the number of more complicated and slower tuple collections. When we put an operator which generates the negative tuple on a higher position in a query production plan, the lower number of operators became strict nonmonotonic. When we put the operators fed by the weakest nonmonotonic operators at lower position in a query production plan, we reduce the number of tuple collections that process the weak nonmonotonic streams. The above rules are added to a rule optimizer which reorders query operators in the following ways:

- Selection operators are shifted to the lowest acceptable positions in a query,
- Window operators are lifted to the highest acceptable positions in a query.

The created nearly-optimizer is aware of operator monotonicity; it reorders query operators in such a way that a query production plan has less complicated and slower tuple collections. In comparison to the nearly-optimizers based on statistics such as stream rates and operator selectivity, our optimizer identifies the complexity of data collection management.

It is worth noticing that the operator monotonicity, which is a static operator property, enriches the description of a stream query (DAG) in such a way that the introduced nearly-optimizer is able to identify sub-DAGs which are of benefit to further query optimization. Next, those sub-DAGs can be dynamically optimized with the help of two algorithms, adaptive caching (Babu et al. 2005) and data synopses (Arasu et al. 2006). Another area of further optimizer research is creation of composed operators which consist of basic physical operators. Suppose that our optimizer identifies a group of strictly nonmonotonic operators; then this knowledge can be used to create a composed operator which shares the tuple collections between physical operators in order not to duplicate nonmonotonic collections.

9.3 Query Language

Many query languages have been proposed to stream databases such as CQL (Arasu et al. 2006), Cayuga (Demers et al. 2007), Esper, and Streaming SQL (Namit et al. 2008; Yijian et al. 2006) which belong to declarative languages. Even though the Object Representation of Query (ORQ) is a commonly used part of SDMS, we have not come across extensions of query languages which use elements of the Object-Oriented (OO) paradigm in order to automatize mapping new DSMS functionalities to the query language. The development of most stream query languages can be named descending, because at the beginning a new syntax of language extension is defined, and then it is implemented in ORQ. In contrast to them, we used elements of OO paradigm in order to invert the development of our query language. Thanks to that, we can extend the object representation of a query, and it is automatically available form the query language level. Moreover, this approach systematizes the way the language evolves. Let us notice that the OO paradigm can also make the query language syntax more confusing when the object representation of a query is complicated.

Now we will specify the *data factories* and the *data collections* which are used in our language presentation. *Data collections* describe the schemes of streams or relations. Moreover, they are accessed by *data factories*. The OO paradigm is used to represent *data factories* as objects which supply methods that transform a data factory into a stream or a relation.

Let us follow the example below. We want to create a tuple-based window on stream S with the size of five tuples. According to our notation, this is expressed by `S{rangeWindow(5)}`, where `S{...}` indicates the *data factory* related to stream S. `rangeWindow(5)` is an object method which creates the tuple-based window with the size of five tuples.

9.3.1 Structure of Query Language

The StreamAPAS users define *units* which represent groups of queries and then use commands `compile`, `run` and `remove` to control the state of those *units* in DSMS. This approach simplifies the management of query resources, because it introduces a higher level of abstraction where we do not have to control each subquery individually. Stream processing applications are defined as an acyclic directed graph whose nodes represent operators and edges represent data transfer. Those operators are created and configured by means of *tasks* which define the production plans of streams (and other data structure in future). The query language syntax allows us to define *tasks* directly by means of methods and indirectly by a syntax similar to SQL/CQL.

StreamAPAS is implemented in Java, and therefore we have decided to allow users to extend the functionality of DSMS by means of packages.

We have defined the following syntax of method call:

```
[<fully qualified class name>.]<method name>
            [::method modifier]([<arg list>])
```

where:

- a fully qualified class name is a class name with a path to the name scope from which the class is referenced. This notation is equivalent to Java notation of fully qualified class name.
- a method modifier is an element which modifies the way how a method call is interpreted. Currently the system defines *task* modifier.

When we call a method which is defined in a current local scope, then we do not indicate a *fully qualified class name*. We have to write a *fully qualified class name* when we call a static method (class method). The *task* modifier is used when a method creates an object *task*. It is necessary for the compiler to arrange the hierarchy of task name scopes correctly.

Example 9.1 We want to collect the times of result latencies measured for the *distinct* operator.

```
test run
  begin
    Benchmark.RandomStream::task("I")
    S{Set.distinct(I{}, "valL")}
    Gui.showAndRegisterLatency::task(S{},"out.txt")
end;
```

This example is solved by a *unit* named `test`. It consists of a *task* which generates random stream I. Then stream S is defined which is a result of the `distinct` operator. Next, S is visualized, and the result latencies are saved to file "out.txt." Let us notice that *task* objects are created by the class methods `RandomStream` and `showAndRegisterLatency`. The object representing the operator `distinct` in ORQ is also created by class method `Set.distinct`. Figure 9.3 shows the hierarchy of *units* and *tasks* for Example 9.1.

The queries below show the full syntax of *unit*. Additionally, the language offers a shortcut which allows users to define a *unit* and run it in one call as it was described in the previous example.

```
//compilation and registration
test compile
  begin

  end;

//starting query
Test run;

//removing query
test delete;
```

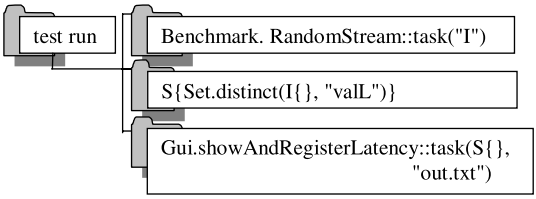


Fig. 9.3 The query division into unit and tasks for Example 9.1

Usually the data processing system consists of a number of streams which cooperate only at the level of a given *unit* or *task*; so this is not necessary to make them accessible from different levels. Therefore *units* and *tasks* create a tree in which each node has an associated scope of *data factory* names.

Figure 9.3 illustrates an example structure of name scopes. We have implemented the following management of name scopes. Let us assume that we refer to data factory I by the query language. This means that the compiler searches for I declaration, at first, in the current local scope and then recursively in the parent scopes. This mechanism is enriched with shortcut objects which reside in a local name scope and point at *data factories* defined in other name scopes. Those shortcuts can be used to refer to *data factories* declared in higher levels of name scope hierarchy by single name.

The hierarchy of name scopes needs each data factory to be a qualified name which consists of a data factory name and the path which leads from a *unit* node to the name scope where the data factory is defined. In consequence, each name scope node has to be uniquely identified by name. Let us notice that users do not have to use qualified names directly to refer to *data factories* declared in children scopes because those *data factories* are reachable through shortcuts. Thanks to that, the compiler can automatically assign unique names to *task* nodes, whereas the name of the *unit* node is specified by user.

In order to create shortcuts, we have defined the *public* modificator to *data factories*. This modificator orders the compiler for the creation of a shortcut to a *given data factory* into the parent scope. In Example 9.1, modifier *public* is assigned to stream I. Thanks to that, this stream is reachable from other tasks inside the *unit*.

9.3.2 Syntax of Unit and Task

Data stream databases are the subject of intensive research which covers new schedulers, stream operators, indexing structures, and DSMS architectures. Therefore we have decided to develop a language which requires little effort to extend it in order to test those new functionalities.

Adding new functionalities to the SQL language is connected with reediting the language syntax. Our aim is to reduce the necessity of language syntax changes in

such situations. In consequence, it will be easier to adapt it to the changing DSMS environment.

Each *unit* and *task* can be mapped to an object which belongs to ORQ. Let us notice that those objects can be created and manipulated directly by the query language if only a query language supports the OO paradigm. In consequence, changes of ORQ would be automatically mapped onto the query language. In order to implement this, each element in the language has its own name scope which contains the names of object methods. In the current system, we distinguish three areas which have defined object methods. The name scope of a *unit*'s methods which is a body of *units* between `start` and `end`. The name scopes of the *data factories* and the attribute tree nodes which are delimited by `{}` and `[]`. The implementation details of how object methods are made accessible from the query language are discussed in Sect. 9.3.4.

Example 9.2 We want a *unit* to use scheduler `XYZ`. This scheduler is defined in the `scheduler` package, and the method which configures `XYZ` is named `BasicCfg`.

```
test run
  begin
    setScheduler(shedulers.XYZ.BasicCfg())
    ...
  end;
```

In the example above, there is a class method called `BasicCfg` which creates an object that represents the configuration of scheduler `XYZ`. Then this configuration is passed to the *unit* by calling object method `setScheduler`.

Summing up, this example illustrates the power of expressiveness when we allow users to create some parts of ORQ directly from the query language. Let us notice that the language syntax is static, and only the contents of the name scope are subject to change.

9.3.3 Attribute Tree

Stream data bases can be classified as data warehouses which are intended to calculate nearly real response time. In such applications, the data organized hierarchically facilitates the manipulation and interpretation of results. Hierarchical data can be used to group attributes thematically. For instance, a car can be described by a unique identifier and a node which represent its position. Then the position node can consist of attributes x, y. Hierarchical data is also useful in reflecting the organization of aggregates. SQL and CQL languages define relation or stream schemas as a list of attributes. When we want to create a data schema similar to a hierarchical data structure by means of SQL, it is necessary to define new custom data types.

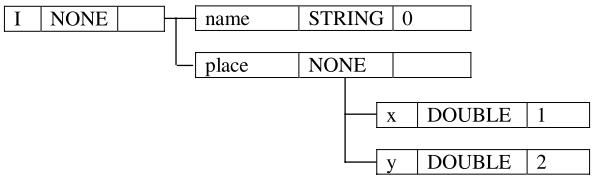


Fig. 9.4 Sample attributes tree

Contrary to the SQL approach, the attribute tree allows us to create a hierarchical data structure as a composition of simple expressions.

The attribute tree is represented by nested lists of nodes which have names and declarations of the node value type. These declarations consist of object type declaration and index value indicating the localization of object values in tuples. The label of the attribute-tree root represents the name of the data collection (e.g., stream). Not all the nodes contain values; those empty nodes are of `none` type. Such nodes are useful in defining groups of attributes. For example, the *place* node in Fig. 9.4 represents the group of position attributes.

If we define a query expression which reads attribute `I.place.x` (Fig. 9.4), the compiler gets the *index* attribute from the `I.place.x` node, and then this *index* = *1* value is used to get the attribute values form the tuples.

The basic syntax of tree logic formulas is presented below. It results from syntax proposed in Cardelli and Ghelli (2001).

η ::= label expression
  \$*name* search for a node in namespace of data collections
  name search for a node in current namespace
A, B ::= formula
  true subtree of current node
  $\eta[A]$ location of current node
  A, B composition of formulas

The composition of formulas allows us to create a new data structure from primitive formulas. The query language is able to check the equivalence of attribute tree structure types, thanks to that subtrees of attribute trees and single nodes can be used as function operands. Suppose that we want to use the subtree of the `place` node as an operand. In this situation we write the name `$I.place.x[true]`.To simplify the syntax, e.g., `O{z[x = 1.1]}`, the series of brackets `[]` can be replaced by `O{z.x = 1.1,...}`.This syntax is known in OO languages as "dot syntax." At the time of the declaration of an attribute-tree node, the node has no data type. The first assignment operation that is called on this node determines its type. When we assign a value to an attribute-tree node that has a predefined data type, an attempt is made to cast the value type on the data type of this node. If there is no defined cast for a given data type, the compilation error will be sent. When we refer to the data collection object, we have to use braces `{}`. This syntax allows the compiler to distinguish whether we operate on tuples or on data collections.

In the example below, we show how to declare the stream which consists of the sum of place coordinates and the subtree of the `place` node.

```
O{z = $I.place.x + $I.place.y,
  $I.place[true]}
```

At the beginning of this formula, we declare `O` stream. Then we declare the stream as a composition of attributes `z` and `place`. Next, we define how the values of those attributes are calculated. The formula not only defines the attribute tree but also the *data collection* object.

The use of `$` depends on whether a fully qualified name is needed. At the beginning, the name scope points to a catalog of available data collections, and therefore the name of data collection is not preceded by `$`.When we are inside the declaration of a data collection, the other collections and their attributes must be fully qualified.

Summing up, when the data collection schemas are well designed, long lists of attributes which are met in SQL can be replaced with a shorter one consisting of attribute tree nodes. The language which is defined to manipulate attribute trees joins two functionalities. This enables the user to define the *data collection* schema and calculation plans together.

9.3.4 Functions

The limited number of available data types diminishes any system usability. To avoid this, we have implemented abstract types in the StreamAPAS query language. When a new data type is needed, we have to add new custom functions that create and operate this new object type. For instance, the `RandomStream` function creates object that implements the `task` interface.

Our compiler has been implemented in Java, so we have decided to use a reflective programming paradigm. In order to add a new function to the query language, the user defines this function in a java class. Then *classpath* to this class has to be added to the library manager of the compiler. The reflection allows us to search methods by their names and their argument lists inside *.class* file. Let us notice that arithmetic functions are called when streams are processed, whereas the `RandomStream` function is called during the compilation phase. Therefore, the compiler needs additional information on the role of the method. We associate this information with method by means of the annotation mechanism. In default, all methods are recognized as arithmetic functions, when a given method has to be called during the compilation phase, we annotated method with `@MModificator(mode = MModificator.CUSTOM_OP_BASE)`. This mechanism allows us to integrate specialized function modifiers defined in the query language with the Java language code.

Reflection gives unlimited access to all the methods defined inside classes. This access policy is unacceptable because some methods should be achievable only from the Java code not from the query language. In order to define a new access policy,

we use the annotation mechanism. In default, all the methods are visible to the query language. When we want to hide all the methods of a given class, we annotate the class with `@CModificator(mode = CModificator.HIDEMEMBERS)`. When some method should be accessible to the query language, the role in the query language have to be assigned directly.

Summing up, it is not a complicated task to define a new custom function and add it to the query language because we need the syntax of Java language only. Moreover, this approach simplifies testing new methods. The java code bellow shows sample declaration of the `StreamUniformRandom` class function.

```
@CModificator(
 mode = CModificator.HIDEMEMBERS)
public class benchmark {

@MModificator(
 mode=MModificator.CUSTOM_OP_BASE...)
public static OperatorBase
                                StreamUniformRandom(...
    ...
```

Because data collections in the query language are also represented as objects, a data collection can have methods that are accessible at the query language level. Similarly to class functions, the object functions are annotated by `@MModificator( mode = MModificator.CUSTOM_OP_BASE)`. An example of object function is `rangeWindow`.

9.4 Linear Road Benchmark

The simplified version of the original linear road benchmark will be used to compare syntaxes of another stream query language named CQL with our language.

The linear road application computes the fee for each vehicle on the motorway individually, in such a way that vehicles visiting congested segments of the motorway pay a higher fee. In consequence, the traffic on the motorway is balanced, because drivers chose other routes so as to minimize the fee to be paid. This benchmark assumes that each vehicle has a sensor which transmits the values of speed and location. Then this information is transferred through a sensor system to the central server which updates the rates and the fees. Next, the updated individual fee and rates for using the motorway segments are forwarded back to vehicle's sensors. A detailed description of the linear road benchmark is available in Arasu et al. (2004).

Figure 9.5 describes the elements of the motorway system. There are L lanes which are numbered $0, \ldots, L-1$. Each lane is 100 miles long and runs east–west. The motorway is divided into 100 segments whose boundaries have entrance and exit ramps. Vehicles on the motorway transmit their positions and speeds every

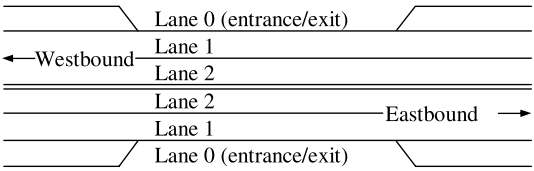


Fig. 9.5 A sample segment of the Linear Road motorway

30 seconds. The position is defined as a lane number, a direction (east/west), and a distance measured from the left end of the motorway.

Vehicles pay a fee when they go through congested segments. A segment is considered congested when the average speed of all the vehicles in this segment during the last 5 minutes is lower than 40 MPH. The fee rates are calculated according to the following formula: $baseFeeRate * (numVehicule - 150)^2$.

In the remaining part of the chapter, we will assume that the linear road application generates stream *SegSpeedStr*(*vehiculeId, speed, segNo, dir, hwy*). The attribute *vehiculeId* identifies a vehicle, *speed* is the speed in MPH, *segNo* denotes segment where vehicle is, *dir* denotes the direction (east/west), and *hwy* denotes the motorway number.

9.4.1 CQL

CQL originates from SQL which was extended by two language syntaxes that correspond to stream-to-relation and relation-to-stream operators. In consequence, queries in CQL are easy to express by users familiar with SQL. In order to illustrate the CQL syntax, we will show examples of queries which resolve parts of the liner road benchmark.

Example 9.3 We want to know which vehicles are active. A vehicle is active when it has transmitted its position during the last 30 seconds.

```
Select Istream(distinct vehiculeId)
From SegSpeedStr[Range 30 Seconds]
```

The above query illustrates the usage of all the operator classes. First, a stream-to-relation operator represented by the time-sliding window is applied. Then, the output relation is processed by two relation-to-relation operators, projection and duplicate elimination, respectively. Finally, the result relation is converted into a stream with a relation-to-stream operator represented by `Istream`.

In order to reduce common operators in expressions, CQL specifies the following syntactic shortcuts. If a query omits the specification of a stream-to-relation operator and the semantic needs a relation, then the compiler applies window `S[Range Unbounded]`. The compiler also inserts `Istream` after the root operator of Q when the result of Q is monotonic and a relation-to-stream operator is not specified. Those rules applied to the previous example result in the following query:

```
Select distinct vehiculeId
From SegSpeedStr[Range 30 Seconds]
```

Other similarities between CQL (Arasu et al. 2006) and SQL are illustrated by the queries below.

Example 9.4 We want to create a relation which contains all the segments containing active vehicles.

```
Select distinct L.vehiculeId, L.segNo, L.dir, L.hwy
From SegSpeedStr[Range 30 Seconds] as A,
     SegSeedStr[Partition by vehiculeId Row 1] as L
Where A.vehiculeId = L.vehiculeId
```

Example 9.5 We want to create a relation which contains all the congested segments.

```
Select segNo, dir, hwy
From SegSpeedStr[Range 5 Minutes]
Group By segNo, dir, hwy
Having Avg(speed) < 40
```

Example 9.6 We want to calculate the number of vehicles in segments.

```
Select segNo, dir, hwy,
       count(vehiculeId) as numVehicles
From ActiveVehiculeSegRel
Group by segNo, dir, hwy
```

9.4.2 Example Queries in StreamAPAS

The query bellow shows how Example 9.3 can be expressed in our language.

```
select result{$SegSpeedStr.vehiculeId}
where SegSpeedStr{slideWindow(30 000)}

resultDist{Set.distinct(result{}, "vehiculeId")}
```

At first, a time-based window is declared with the size of 30 seconds. Then the result is saved to stream *result*. In the next sub query, a `distinct` operator is defined which saves *result* to stream *resultDist*. It is worth noticing that the `distinct` operator is created by calling the class method: `Set.distinct`. In contrast to that, the time-based window is created with the use of the object method `slideWindow`. This method is declared in the name scope of the *data factory* which represents stream *SegSpeedStr*.

The following queries show how Examples 9.4–9.6 defined in CQL can be expressed in our language.

Example 9.7

```
Select tmp{$L.vehiculeId, $L.segNo, $L.dir, $L.hwy}
From L{SegSpeedStr{}}
Where SegSpeedStr{slideWindow(30 000)},
            L{partitionedWindow(1,"vehiculeId"),
            SegSpeedStr.vehiculeId = L.vehiculeId

ActiveVehicleSegRel{Set.distinct(tmp{},
                                  "vehiculeId")}
```

Example 9.8

```
select CongestedSegRel{$SegSpeedStr.segNo.segNo,
                       $SegSpeedStr.segNo.dir,
                       $SegSpeedStr.segNo.hwy}
where SegSpeedStr{slideWindow(300 000)}
group by SegSpeedStr.segNo, SegSpeedStr.dir,
         SegSpeedStr.hwy
having Agg.sum($SegSpeedStr.speed) < 40
```

Example 9.9

```
Select SegVolRel($ActiveVehiculeSegRel.segNo,
                 $ActiveVehiculeSegRel.dir,
                 $ActiveVehiculeSegRel.hwy,
                 numVehicles = Agg.count())
group by ActiveVehiculeSegRel.segNo,
         ActiveVehiculeSegRel.dir,
         ActiveVehiculeSegRel.hwy
```

9.5 Further Work

Development of data warehouses in streaming environments is a promising area of new applications, and this constitutes the basis of our further work. In the chapter we have illustrated the query language. Its syntax is motivated by our previous tests with indexing structures designed for spatio-temporal data.

Let us notice that any indexing structure can be easily represented as a new *task*. This process requires us to load package with new functionalities. Then we can use its class methods to define the specification of an indexing structure, in a way similar to the one shown in Example 9.2 and the examples below.

Example 9.10 We want to create index *traffic* which is an R-tree supplied with streaming information of vehicle positions transmitted by stream *I*. The query bellow shows the potential beginning of this solution.

```
gis.Rtree::task("traffic", {$I.point[true]},...)
```

Example 9.11 Then we want to use index *traffic* to find the nearest five vehicles which may be sent to accidents. Let those accidents be notified by stream *help* which transfers tuples with accident positions. The query below shows the *data factory* as an interface used to define operations on the index structure.

```
select result{$traffic{}}
where traffic{kNN($help{})};
```

Example 9.12 The syntax of the query language associates a tree with a *data factory*. This tree represents dimensions and subdimensions. Besides, each node has its own scope of function names. Thanks to that we can look at defining queries upon indexes in a way similar to the one used in data warehouses. In order to define the operator which extracts data from a *data factory*, the user calls the methods of this node tree. Let us assume that index *traffic* is an aggregate tree and we want to calculate the number of vehicles falling into a given area. The example below illustrates how it could be expressed with the use of the *data factory* syntax.

```
select result{$traffic{}}
where traffic{contain ($areas{}), measures[sum()]};
```

9.6 Related Work

There are a lot of stream processing research projects such as STREAM (Motwani et al. 2003), Telegraph (Shah et al. 2001), TelegraphCQ (Sirish et al. 2003), and Aurora & Borealis (Balazinska 2006). There are as many stream query languages proposals as many projects are carried. We can divide those languages into three categories. In one approach, each operator connection is defined directly in a text file or graphically (Balazinska 2006). In a procedural language, a user defines query as a loops that are fed with tuples from stream collections. The most promising role plays declarative languages (Motwani et al. 2003) because their users may be not aware of the physical realization (e.g., chosen algorithms). There, the query optimizer is responsible for translating a task defined on an abstract level into a physical realization.

Stream processing becomes popular in online analysis based on aggregates defined over different windows, data sequence, prediction, and many more. Undoubtedly, those functionalities are closer to data warehouse tools rather than traditional database operations. Therefore, we not only consider SQL syntax as a basis for further research but also CQL and MDX gain our interest. There are a lot of propositions of SQL syntax extensions which usually address a particular class of problems, for instance, time sequence analysis (Ali et al. 2005).

9.7 Conclusion

In the developed language, we test the combination of the Object-Oriented paradigm and declarative languages including the languages designed for multidimensional data analysis. The proposed query language is not a complete solution; however, it has a systematized approach to functionality extensions so that the implementation of the syntactic analyzer is easier.

In the chapter, we describe the impact of query logic representation on an optimization phase. The proposed query nearly optimizer reduces the number of strict nonmonotonic operators as a result of which the system transmits the lower number of negative tuples. Moreover, we show how the information on monotonicity of operator can be used to accelerate tuple collections governed by physical operators.

Finally, we introduce the attribute tree which represents spatial and analytical data in a more convenient way, because it allows the user to define expressions on single attribute or a group of attributes.

The current stream processing engine supports only stream data collections. It is a subject of our further research to add relations and more sophisticated index structures into the system.

References

Abadi, D.J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., Zdonik, S.: Aurora: a new model and architecture for data stream management. The VLDB Journal **12**(2), 120–139 (2003)

Ali, M.H., Aref, W.G., Bose, R., Elmagarmid, A.K., Helal, A., Kamel, I., Mokbel, M.F.: NILE-PDT: a phenomenon detection and tracking framework for data stream management systems. In: VLDB '05: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 1295–1298. VLDB Endowment (2005)

Arasu, A., Cherniack, M., Galvez, E.F., Maier, D., Maskey, A., Ryvkina, E., Stonebraker, M., Tibbetts, R.: Linear road: A stream data management benchmark. In: M.A. Nascimento, M.T. Özsu, D. Kossmann, R.J. Miller, J.A. Blakeley, K.B. Schiefer (eds.) VLDB, pp. 480–491. Morgan Kaufmann, San Mateo (2004)

Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. The VLDB Journal **15**(2), 121–142 (2006)

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: PODS '02: Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 1–16. ACM, New York (2002)

Babu, S., Munagala, K., Widom, J., Motwani, R.: Adaptive caching for continuous queries. In: ICDE '05: Proceedings of the 21st International Conference on Data Engineering, pp. 118–129. IEEE Computer Society, Washington (2005)

Balazinska, M.: Fault-tolerance and load management in a distributed stream processing system. Ph.D. thesis, Cambridge, MA, USA (2006)

Cardelli, L., Ghelli, G.: A query language based on the ambient logic. In: ESOP '01: Proceedings of the 10th European Symposium on Programming Languages and Systems, pp. 1–22. Springer, London (2001)

Demers, A.J., Gehrke, J., Panda, B., Riedewald, M., Sharma, V., White, W.M.: Cayuga: A general purpose event monitoring system. In: CIDR, pp. 412–422 (2007)

Ghanem, T.M., Hammad, M.A., Mokbel, M.F., Aref, W.G., Elmagarmid, A.K.: Query processing using negative tuples in stream query engines. Tech. Rep. 04-040, Purdue University (2005)

Golab, L.: Sliding window query processing over data streams. Ph.D. thesis, University of Waterloo (2006)

Krämer, J.: Continuous queries over data streams semantics and implementation. Ph.D. thesis, Philipps-Universität Marburg (2007)

Krämer, J., Seeger, B.: A temporal foundation for continuous queries over data streams. In: COMAD, pp. 70–82 (2005)

Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., Manku, G., Olston, C., Rosenstein, J., Varma, R.: Query processing, resource management, and approximation in a data stream management system. In: CIDR, pp. 245–256. CIDR (2003)

Namit, J., Shailendra, M., Anand, S., Johannes, G., Jennifer, W., Hari, B., Çetintemel, U., Mitch, C., Richard, T., Stan, Z.: Towards a Streaming SQL Standard. pp. 1379–1390. VLDB Endowment (2008)

Shah, M.A., Franklin, M.J., Madden, S., Hellerstein, J.M.: Java support for data-intensive systems: experiences building the telegraph dataflow system. SIGMOD Record **30**(4), 103–114 (2001)

Sirish, C., Owen, C., Amol, D., Wei, H., Sailesh, K., Samuel, M., Vijayshankar, R., Frederick, R.: TelegraphCQ: Continuous dataflow processing for an uncertain world. In: CIDR (2003)

Tucker: Punctuated data streams. Ph.D. thesis, OGI School of Science & Technology At Oregon Heath (2005)

Yan-Nei, L., Haixun, W., Zaniolo, C.: Query languages and data models for database sequences and data streams. In: Proceedings of the VLDB International Conference of Very Large Data Bases, pp. 492–503 (2004)

Yijian, B., Hetal, T., Haixun, W., Chang, L., Zaniolo, C.: A data stream language and system designed for power and extensibility. In: CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 337–346. ACM, New York (2006)

Chapter 10
Agent-Supported Programming of Multicore Computing Systems

Sabri Pllana, Siegfried Benkner,
Eduard Mehofer, Lasse Natvig, and Fatos Xhafa

Summary In this chapter we argue that an intelligent program development environment that proactively supports the user helps a mainstream programmer to overcome the difficulties of programming multicore computing systems. We propose a programming environment based on intelligent software agents that enables users to work at a high level of abstraction while automating low-level implementation activities. The programming environment supports program composition in a model-driven development fashion using parallel building blocks and proactively assists the user during major phases of program development and performance tuning. We highlight the potential benefits of using such a programming environment with usage scenarios. An experiment with a parallel building block on a Sun UltraSPARC T2 Plus processor shows how the system may assist the programmer in achieving performance improvements.

S. Pllana (✉) · S. Benkner · E. Mehofer
Department of Scientific Computing, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria
e-mail: pllana@par.univie.ac.at

S. Benkner
e-mail: sigi@par.univie.ac.at

E. Mehofer
e-mail: mehofer@par.univie.ac.at

L. Natvig
Department of Computer and Information Science, NTNU, Sem Saelands vei 9, 7491 Trondheim, Norway
e-mail: Lasse.Natvig@idi.ntnu.no

F. Xhafa
Department of Computer Science and Information Systems, Birkbeck, University of London, London, UK
e-mail: fatos@dcs.bbk.ac.uk

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_10,

10.1 Introduction

While multicore processors alleviate several problems that are related to single-core processors, known as *memory wall*, *power wall*, or *instruction-level parallelism wall,* they raise the issue of the *programmability wall*. On the one hand, program development for multicore processors, especially for heterogeneous multicore processors, is significantly more complex than for single-core processors. On the other hand, programmers have been traditionally trained for the development of sequential programs, and only a small percentage of them have experience with parallel programming.

Additionally, there is a portability problem. In the past programmers could trust that compilers succeeded to pass the increased computing power of next processor generations without high porting effort. This was due to relatively homogeneous processor designs even from different hardware vendors with instruction level parallelism (ILP) supported at hardware level. The architectural change to multicore processors, however, affects the programmer in several ways. On the one hand, thread level parallelism (TLP) must be exploited effectively and efficiently. In general, this cannot be done automatically by a compilation system but requires assistance by the programmer. On the other hand, multicore architectures differ significantly requiring that applications must be adapted to the various platforms.

While in the past only a relatively small group of programmers interested in High Performance Computing (HPC) was concerned with the parallel programming issues, the situation has changed dramatically with the appearance of multicore processors on commonly used computing systems. Traditionally parallel programs in HPC community have been developed by *heroic programmers*[1] using a simple text editor as programming environment, programming at a low-level of abstraction, and doing manual performance optimization. It is expected that with the pervasiveness of multicore processors parallel programming will become mainstream, but it cannot be expected that a mainstream programmer will like to become an HPC hero.

In this chapter we argue that the programming productivity of multicore[2] systems is increased if an intelligent programming environment would be available that (1) enables the programmer to work during the process of program development at a higher level of abstraction using domain-specific modeling languages in a model-driven development fashion and (2) provides context-specific knowledge and performs iterative time-consuming tasks involved in program development in a semi automatic/autonomic manner (for instance, performance tuning). We propose a parallel programming methodology that combines model-driven and agent-supported program development with the use of high-level parallel building blocks. The goal is to increase programming productivity without restricting flexibility and creativity, allowing the programmer to fully use his/her intellectual capacity for software

[1] Andrea: “Unhappy is the land that breeds no hero." Galileo: “No, Andrea: Unhappy is the land that needs a hero." – Bertolt Brecht in *Life of Galileo*.

[2] Although some authors have introduced the term *many-core* to denote multicore systems with many cores (i.e., 100 or more), we will stick to the more established term multicore. We do not see a need to make a distinction between multi and many.

design at model-level. Although software development is considered to be an art, we anticipate that there are many implementation activities that can be performed more automatically/autonomically.

The rest of this chapter is organized as follows. An overview of the recent developments in parallel computing systems is given in Sect. 10.2. Section 10.3 describes our vision for programming of multicore computing systems. We illustrate our approach experimentally in Sect. 10.4. Section 10.5 reviews the state-of-the-art in programming multicore computing systems. We conclude the paper with a summary and future work in Sect. 10.6.

10.2 Recent Developments in Parallel Computing Systems

In this section we provide an overview of the recent developments in parallel computing systems focusing on (1) parallel and distributed programming, (2) compilation techniques, and (3) multicore architectures.

10.2.1 Parallel and Distributed Programming

The dominating programming paradigm for parallel systems is based upon standard sequential programming languages, augmented with message passing constructs. In particular the standardized Message-Passing Interface (MPI) is widely used for parallel programming. In this low-level model the user has to deal with all aspects of parallelization, distribution of data and work to processors, and communication and synchronization by means of explicit message passing operations. This leads to high cost for software and error-prone programs that are difficult to write, reuse, and maintain. On smaller-scale Symmetric Multiprocessing (SMP) systems, the use of low-level multithreading libraries, such as POSIX threads, faces similar problems.

Despite significant research efforts, automatic parallelization, i.e., taking a serial program written in a mainstream language and automatically generating an executable program capable of taking advantage of parallel hardware, has not been successful, either for shared-memory or distributed-memory systems, and remains an elusive goal. Data-parallel languages such as High Performance Fortran (High Performance Fortran Forum 1997) provide high-level support for controlling locality by associating distributions with arrays while delegating the generation of explicit message passing code to the compiler. Despite some successes, these languages have not been broadly accepted. OpenMP raises the level of abstraction for multithreaded programming but lacks constructs for controlling data locality and is thus constrained to small-scale, homogeneous shared-memory systems. Partitioned Global Address Space (PGAS) languages like Co-Array Fortran (Numerich and Reid 1998), Unified Parallel C (UPC Consortium 2005), and Titanium (Yelick et al. 1998) have not succeeded in breaking the dominance of low-level message-passing and thread programming either. More recent efforts in the context of the US

High Productivity Computing Systems (HPCS) program, characterized by putting an increased focus on programmability and not just performance, have resulted in the definition of new languages including Fortress (Allen et al. 2008), Chapel (Cray 2008), and X10 (Charles et al. 2006) that address the challenges of programming large-scale (PetaFlop/s) systems. So far, however, none of these languages is seeing user-uptake beyond a rather limited research community.

With the emergence of multicore computing systems consisting of hundreds of processor cores in the near future, the challenge of parallel programming will not be restricted to the HPC community any more but will extend to the broad software industry. Single-chip multicore systems will provide performance levels accommodating applications previously only available on clusters and parallel systems and will give rise to new exciting applications in the embedded and mobile computing domains. However, since emerging heterogeneous multi-core systems provide different types of cores including general-purpose cores, GPU-like cores and specialized accelerators, the challenges of efficient programming and of achieving portability across different architectures and architecture generations will be aggravated compared to traditional parallel systems.

A variety of technologies and tools for programming heterogeneous multicore architectures have been made available recently by hardware vendors including TBB (Threading Building Blocks) (Intel Corp. 2009), CUDA (NVIDIA Corp. CUDA Zone 2009), Cell SDK (IBM 2008), and others. All these technologies are characterized by an extremely low level of abstraction, forcing programmers to take into account a myriad of architecture details, usually beyond the capabilities of average users. Programs relying on these technologies are not portable to other multicore architectures. Although recently with OpenCL (Khronos OpenCL Working Group 2009) and MCAPI two proposals for standardizing parallel programming of heterogeneous multicore systems have been announced, these standards primarily address portability issues but are still at a very low-level of abstraction.

Some of the programming challenges that are posed by emerging heterogeneous manycore architectures are similar to those faced in programming large-scale distributed computing systems (Hall et al. 2008), which are often referred to as Grid (Foster and Kesselman 2003). In the context of Grid computing systems, application developer usually should deal with diversity of programming models and languages, computation partitioning, data flow and dependencies among the application components, legacy code and third-party software components, and diversity of computational resources. Furthermore, application components are usually executed by heterogeneous computational resources that are not known prior to execution. Usually, Grid applications are expressed as workflows, and a workflow planning/optimization system maps workflow components to available resources for execution (Taylor et al. 2006).

10.2.2 Compilation Techniques

Heterogeneous multicore parallel architectures are typically shipped with ANSI C compilers for each type of core, and by default it is the programmer's responsibility to write separate programs for each core, plus glue code so that these programs can interact. Advanced compilation techniques aiming to avoid this manual effort tend to be tied into parallel programming systems, where language extensions are used to indicate where to parallelize and to demarcate code into portions for which distinct compilation strategies are appropriate.

Sequoia (Fatahalian et al. 2006) (Stanford University) provides a high-level abstraction for distributing tasks over multiprocessor systems with hierarchical memory, implemented for the Cell BE processor and for multicore x86 clusters. Efficient scheduling of data-movement is eased by requiring the programmer to specify a priori the working set required by a parallel task. Distribution of data across the memory hierarchy is made efficient via a sequence of compiler optimizations (Knight et al. 2009) and via so-called tunable parameters which may be adjusted by the user. A similar mechanism for specifying data usage upfront is used by CellSs (Barcelona Supercomputer Center) (Perez et al. 2007). PetaBricks provides an autotuning compiler and a programming language that can express multiple algorithms for solving a specific problem and exposes algorithmic choices to compiler (Ansel et al. 2009).

OpenMP (Chandra et al. 2000) has been applied to heterogeneous multicore with an implementation for the Cell processor. The IBM XLC compiler (IBM XL C/C++ 2009) implements a large part of the standard, using a sophisticated suite of optimizations targeting both the PPE and SPE cores (Eichenberger et al. 2006). The Codeplay Sieve C++ language (Lindley 2007) uses the concept of delayed side-effects to ease dependence analysis, making automatic parallelization of C++ more tractable for C++ software. The Sieve system exploits parallelism via speculative execution and provides mechanisms to support common parallel patterns (Donaldson et al. 2007). Sieve C++ has been shown to be portable across homogeneous and heterogeneous architectures, with implementations for multicore x86, the Ageia PhysX (AGEIA Technologies 2009) accelerator card, and the Cell processor.

Specifying data movement explicitly via intent qualifiers, as in Sequoia and CellSs, is suitable for HPC applications over regular data sets but comes at the expense of expressiveness. On the other hand, while programming architectures like Cell via OpenMP or Sieve C++ is more flexible, since the data set required by a task is implicit in the way data is manipulated, programs in these languages are harder to optimize. Recent research into decoupled access/execute specifications (Howes et al. 2009) aims to provide the best of both worlds, by decoupling the execution of a kernel from its access pattern, via programmer-specified access functions. Initial experiments with the access/execute model involve predefined transformations based on common data access patterns; in principle these transformations can be automated via compiler support based on the polyhedral model (Pouchet et al. 2007).

Recent advances in programmable GPU architectures have led to widespread interest in the use of GPUs for scientific programming. Programming GPUs for scientific applications (Owens et al. 2007) has usually been performed using graphics lan-

guage such as OpenGL (Shreiner et al. 2005) and more recently using stream computing languages such as CUDA (NVIDIA Corp. CUDA Zone 2009) and RapidMind (RapidMind corp. 2009). More recently, approaches based on familiar high-level languages such as C and Fortran have been proposed (Bodin and Bihan 2009). These approaches are directives based, either new ones (Bodin and Bihan 2009) or extension of the OpenMP standard (Chandra et al. 2000). Directives specify computation to be offloaded on the GPU. Parallel loop nests are translated into one of the target GPU-specific programming language. These approaches are very new, and there are still many issues related to performance tuning. These high-level approaches are the most promising for heterogeneous multicore since they help to avoid multiprogramming and to allow the maintenance of a unique source code.

10.2.3 Multi-Core Architectures

For the last 30 years, the makers of General Purpose (GP) CPUs have leveraged continuous improvements in silicon process technology along two axes: the ever decreasing feature size (as described by Moore's Law of 1965) allowed building ever more complex logic into their CPUs, and the transistors could be driven at ever lower voltages and higher frequencies. The former has led to the dominance of just a few advanced highly pipelined, multiscalar processor architectures with out of order execution capability; the latter allowed one to raise clock frequency and thereby CPU performance up to about 3 GHz. Application software developers profited enormously from both trends, since performance improvements were expected to happen automatically at the cost of at most a recompilation, and there was no need to port and optimize applications to many different architectures.

Since about 2001, two significant trends are reshaping the whole computer ecosystem: the traditional CPU evolution did hit the now proverbial "power wall," meaning that further increases in clock speed could only be achieved by dissipating disproportionate amounts of power and were therefore no longer feasible, and special purpose architectures start to rival GP CPUs in the performance/power and performance/cost metrics. Intel's and AMD's answer to the "power wall" are the current line of multicore CPUs that derive their performance from up to eight complex, independent execution units (cores), and no longer from increases in clock frequency.

The need for improvements in power and cost effectiveness is also driving a renaissance of architectures that are tailored to special computational models and use massive parallelism to surpass GP CPUs for these—prime examples are NVIDIA's or AMD's GPUs that combine hundreds of very simple compute units into a very powerful SIMD parallel system, and FPGAs that can be field programmed to perform complex data transformations such as en/decryption and media format conversions at extremely high speeds. These systems need a GP host processor to run the OS and most applications and are coupled to it by a bus interconnect.

Today, explicitly parallel multicore CPUs have taken over the market (90% of all Intel CPUs sold in 2008 had multiple cores), and novel, massively parallel accelerators are making significant inroads. This is not restricted to the high-performance segment—laptop computers offload part of the GUI processing to their GPUs (Apple Macbooks), and low-power parallel accelerators are being used for embedded systems (e.g., for media format conversions or software defined radio). In effect, the era of heterogeneous multicore platforms is already on us.

Today's systems exhibit heterogeneity mainly between the host CPU(s) and the accelerator(s): instruction set and performance characteristics are very much different, and there is usually no shared memory between the components. A typical example is a workstation that combines two Intel Nehalem CPUs with one to four NVIDIA Tesla accelerators. Applications need to be aware of the different capabilities of the CPU vs. the GPU cores, must use different methods to write the respective parallel code components, and have to explicitly manage the data transfer between CPUs and GPUs. Running such an application in an efficient way requires a smart runtime system that optimizes scheduling according to data placement, resource (core) availability, and performance.

On the general purpose CPU side, the future will certainly see further growth in the number of cores: in 2006, Intel has demonstrated the 80 core Polaris chip, Sun has announced a 16-core "Rock" UltraSPARC processor for 2009 availability, and Intel has published their "Larrabee" many-core architecture (Seiler et al. 2008) which is to scale to dozens of cores. A common theme here is to forego some of the advanced architecture features (such as out of order execution) to reduce the complexity and die size of each core and use the "headroom" to both introduce high-performance SIMD compute units per core and increase the core count at the same time. These "small and nimble" cores can deliver great performance for applications that are adapted to them (like graphics processing)—they will perform worse than conventional cores for many nonadapted programs. Thus, designers of future CPUs face a difficult decision: should they go the "all small core" approach which will maximize peak performance, stick with the complex cores that run a wide variety of (nonoptimized) codes well, or create heterogeneous CPUs that combine both kind of cores. Today, it is too early to tell which evolution path the dominating CPU vendors will be taking. A GP CPU that combines different cores remains a distinct possibility. In the embedded systems space, MIPS-based systems have gone up to 64 cores already, and ARM is joining the multicore bandwagon.

The accelerator field is evolving very quickly; GPUs push the number of processing elements and the functionality of them at the same time, thus increasing peak performance and extending their reach from pure SIMD data parallel kernels in the direction of task level or functional parallelism. NVIDIA's recent communications and Intel's entry into the graphics market with the Larrabee architecture provide ample evidence here. A second line of evolution concerns the way of connecting host CPUs with the GPU: higher performing bus connections (like PCI Express 3.0), cache coherent interconnects like Intel's QPI and AMD's Hypertransport, and finally the inclusion of graphics processing elements into the host CPU (as announced from Intel for their Nehalem desktop chips). Combined, these trends will make future GPUs much more similar to the CPUs that they compete with and alleviate the

large performance disparity that we see today between local memory and the bus interconnects.

10.3 Intelligent Programming of Multi-Core Systems

In this section we outline our methodology and the corresponding environment for programming multicore systems.

10.3.1 Methodology

Our parallel programming methodology combines model-driven agent-supported program development with the use of high-level *parallel building blocks* (PBB). We propose to address the complexity of programming multicore systems as follows:

- Raise the level of abstraction at which the programmer performs most of the activities during the process of software development, by using a model-driven development approach combined with PBBs;
- Support the programmer during the software development, by using intelligent software agents for providing context-specific knowledge and automation of iterative activities involved in software development and optimization.

10.3.1.1 Model-Driven Development (MDD)

MDD (Model Driven Architecture 2009) is a software development method that advocates to *first model* a program and *then build* the program code. It is inspired by mature engineering disciplines such as *civil engineering*, where before an artifact (for instance, a *bridge*) is built, the corresponding model is first developed. In software engineering the models are usually described graphically using the Unified Modeling Language (UML). The model should preferably describe the program at an abstraction level that is independent from a specific platform. Models may be used to study the functionality, and the performance of the program before the program code for a specific platform is developed. MDD has the potential to reduce software development time and complexity, by using tools for automatic model-to-code transformation and thereby reducing the programmer's effort for manual coding. Since multicore architectures differ significantly from each other, a significant effort is required to adapt (that is, *port*) programs to the various platforms. Since MDD captures the program logic as a platform-independent model, program models remain largely unaffected from the changes in processor architectures. In our previous work we have developed an extension of UML for the domain of performance-oriented parallel/distributed programs (Pllana et al. 2002) and the corresponding tool-support *Teuta* (Pllana et al. 2004). Teuta allows one to build models of parallel programs, enrich them with performance-related information, and generate various textual representations (such as XML or C++).

10.3.1.2 Parallel Building Blocks

The PBBs are inspired from research in programming concepts such as *skeletons* (Alba et al. 2002; Alind et al. 2008; Cole 2004) or *dwarfs* (Asanovic et al. 2006). Basically, PBBs may be thought of as program-independent generic programming units that support software reusability. A set of parameters is used to specify the functionality of a PBB in the context of a certain program. For instance, as parameter may serve the program-specific code (that is, the code that PBB requires to perform the expected functionality in the context of a certain program). PBBs may be implemented, for instance, using *C++ Templates* or *Java Generics*. Parallelism is described within the PBB, and therefore the programmer is not exposed directly to the parallel programming complexity (such as dealing explicitly with the *communication and synchronization* among processing units or *deadlock avoidance*). Commonly various combinations of PBBs may be used for solving a certain problem. In the context of programming environments, PBBs lend themselves to an increased level of automation of various activities such as program transformation, code generation, performance optimization, and resource usage optimization. In our previous work, in the context of MALLBA project (Alba et al. 2002), we have developed a library of parallel skeletons (such as *branch and bound*, *metropolis*, *simulated annealing*, *genetic algorithms*, or *tabu search*) for solving various optimization problems.

10.3.1.3 Intelligent Software Agents

Software agents are programs that are *reactive*, *proactive*, *autonomic*, and *social* (Wooldridge 2002). Software agents that have *learning* and *adapting* abilities are known as *intelligent software agents*. *Reactiveness* indicates the ability to respond adequately to changes in the context in which it operates. A *proactive* program performs activities to achieve a specific goal based on its initiative (it does not wait passively for a request of another entity to perform a certain activity). *Autonomy* indicates the ability to perform activities independently of user intervention in order to achieve a specific goal. *Social* programs are able to communicate and coordinate activities with other programs (that is, agents). A program is considered intelligent if it is able to learn from the previous experience (for instance, via trial-and-error or generalization) and is able to adapt accordingly to the perceived changes in the environment. We have a vision about several intelligent software agents cooperating with each other and the programmer during the process of program development. Our vision is based on the idea that the programming environment should be better at helping the programmer as a more active partner. In our previous work, in the context of the AURORA project (AURORA 2009), we have used intelligent software agents to automate systematic performance analysis for parallel and distributed programs. Although software development is considered to be an art, we anticipate that there are many implementation activities that can be performed more automatically/autonomically using intelligent software agents.

In the following subsection we propose a programming environment for multicore computing systems that uses MDD, PBBs, and intelligent software agents.

10.3.2 Programming Environment

The proposed programming environment comprises a set of intelligent software agents that may help to automate the programming process at several levels. Some agents will advice the composition of programs using PBBs, while others will guide the exploration of different possible parallel strategies, load balancing, and performance optimization (see Fig. 10.1).

The programming environment provides the programmer with information feedback useful in the process of developing a program for a multicore system. This information is collected at several levels, from program composition to information about resource usage (such as the cache behavior) obtained by execution or simulated execution. Also, information is exchanged between the agents at the system level in an automated manner continuously looking for ways of obtaining and improving knowledge about the performance of the program being developed. In this way, a parallel program with good performance can be developed with high programmer productivity.

In what follows in this section we highlight the major program development and tuning phases: (1) high-level program composition, (2) design space exploration, and (3) resource usage optimization.

10.3.2.1 High-level Program Composition

This phase deals with the composition and coordination of PBBs. The granularity of PBBs may range from frequently used programming idioms to larger patterns or dwarfs (Asanovic et al. 2006). High-level descriptors are used to capture the main parallelization aspects of PBBs and serve as interface to agents in the design

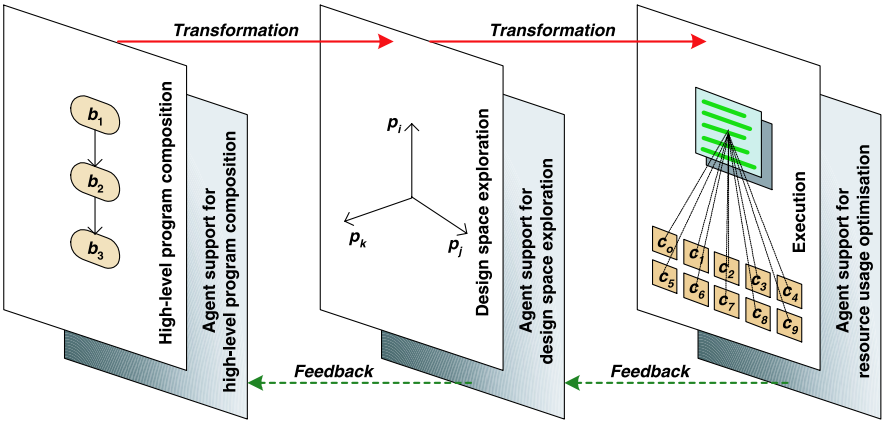


Fig. 10.1 Agent-supported program development. The programming environment comprises multiple intelligent software agents that support program composition, design space exploration, and resource usage optimization

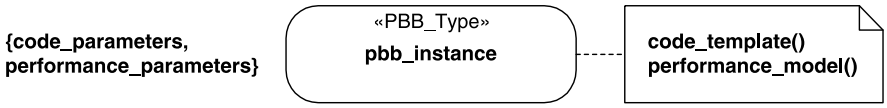


Fig. 10.2 UML representation of a PBB

space exploration phase. The user composes the program graphically using a UML extension for multicore systems.

The UML may be extended by defining new modeling elements, *stereotypes*, based on existing elements (also known as *base classes* or *metaclasses*). Stereotypes are notated by the stereotype name enclosed in guillemets «`Stereotype Name`». Figure 10.2 depicts the graphical representation of a PBB. «`PBB_Type`» indicates the kind of PBB. With a PBB is associated the corresponding *parameterized code and performance model*. Parameters determine the behavior of the PBB instance in the context of a specific program.

The programming environment assists the user proactively during the program composition. For instance, while the user is loading some old BLAS code for some dense linear algebra operations, the *composer agent* interrupts and suggests using the PBB for dense linear algebra tailored for efficient execution on multicore systems. Additionally, it may offer a list of other PBBs that often are used together with this one, also presenting typical compositional patterns in a graphical way.

10.3.2.2 Design Space Exploration

High-level discrete-event simulation is used for rapid model-based performance evaluation of programs, using a hybrid method that combines mathematical modeling with high-level discrete-event simulation (Pllana et al. 2008).

For instance, after the completion of the program composition phase, the programming environment may suggest to the user doing some high-level rapid design space exploration. The estimated performance of various possible program implementations is presented by a *visualization agent*. While the user is studying the graphs and gets some ideas for improvement, the programming environment is also analyzing the results and suggests changing some of the parameters in one of the PBBs (such as the parallelization granularity) and performing some more detailed simulations for getting better knowledge of the performance that can be obtained with different task allocation and scheduling policies.

10.3.2.3 Resource Usage Optimization

Instruction-level simulation is used for more detailed studies of the utilization of shared resources such as shared on-chip memory and off-chip bandwidth. For instance, in Dybdahl et al. (2006) an efficient utilization of the shared cache resources has been found to have great affect on multicore performance. This is inte-

grated with the use of performance counters. A performance monitoring agent provides information about the state of the system (resource characteristics and usage). Instruction-level simulation is time consuming (may take several hours or days) and therefore should run in background. When finished, the findings will be propagated upwards back to the higher-level performance models, as a model calibration process. It is a systematic way of bringing performance information from the execution (or simulated execution) environment back to the development environment. Please note that this kind of optimization is architecture-dependent.

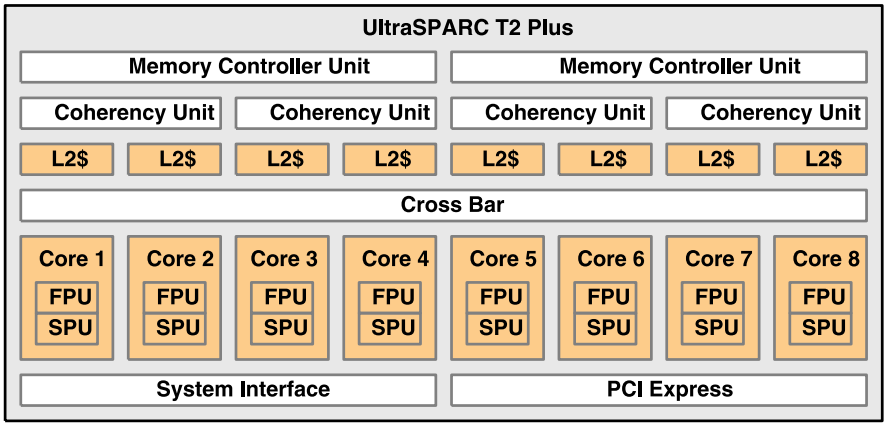
For instance, the user may get hints from the programming environment for changes that will improve performance of the program. The programming environment may offer some detailed simulations at the instruction level and helps the user to select those simulation experiments that are likely to be the most relevant. For instance, if higher-level simulations show that some of the processor cores were waiting for data for long periods, a more detailed study of the on-chip shared memory resources should be done.

10.4 Example

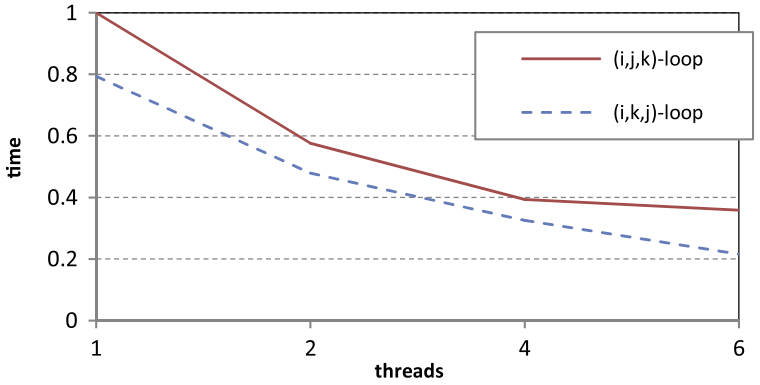
In this section we illustrate how best practices from HPC, combined with agent-based program development, offer new opportunities to obtain efficient solutions.

PBBs allow a programmer to specify various parallelization strategies together with the code and a first guess for individual parameters which are subject to the tuning process. This follows our assumption that only semi-automatic parallelization is reasonable. The programmer specifies the main strategies for parallelizing the code, and the system explores this restricted optimization space to generate efficient code. Two factors back up this approach. First, rich analysis work has been done in the past by the HPC community, including the authors institutions (Vienna Fortran Compilation System, Benkner et al. 1996), which can be reused. Second, in the past the strong emphasis on the target-code performance and manual performance tuning resulted in low programming productivity. The increasing importance of development of economically viable software nowadays reveals opportunities for semiautomatic parallelization, even at the price of achieving lower performance compared to a hand-tuned version.

In our example we use as hardware platform the Sun UltraSPARC T2 Plus, codenamed Niagara-2, multicore processor (shown in Fig. 10.3(a)) which is an SMP extended version of the T2 allowing multiple Chip-level Multithreading (CMT) processors to be used within a single system. The T2 Plus was presented in April 2008 and has up to 8 cores per processor with 8 hardware threads per core resulting in a maximum number of 64 threads per processor or logical CPUs as reported by the operating system. T2 Plus offers only poor support for instruction-level parallelism emphasizing thread-level parallelism. Two integer units are provided per core with four threads sharing one unit, and one FPU is provided per core with all eight threads sharing it. The L1 data cache has 8 KB per core, and the on-chip L2 cache offers 4 MB which are shared between the cores.



(a) Sun UltraSPARC T2 Plus.



(b) Performance improvements.

Fig. 10.3 Processor block diagram and optimization results

In what follows in this section we present an example scenario to illustrate the agent-supported software development cycle. Different forms of PBBs are possible, but in the simplest case a PBB can be some loop nest together with data layout and work distribution annotations. Consider, e.g., an application written in C consisting of a series of PBBs with one of them denoting a floating point matrix–matrix multiplication, i.e., `C[i,j] = C[i,j] + A[i,k] * B[k,j]` with loop nest (i, j, k). As parallelization strategy, the programmer specifies that the elements of result matrix C should be assigned to processor cores in a row-wise manner and calculated by them. Since the target architecture is a Sun T2 Plus with 8 cores and 8 FPUs, the programmer specifies that the rows shall be assigned to 8 threads.

When submitted to the *design space exploration agent* and its analysis framework (cf. Benkner et al. 1996; Benkner 1999), the framework detects poor spatial cache locality and performs loop interchange resulting in loop nest (i, k, j). Then the code is split up in 8 threads, as suggested by the programmer, assigned to the 8 cores

of T2 Plus, and executed. The monitoring component of the *resource usage agent* reveals low memory bandwidth utilization and low FPU utilization for this PBB and reports this feedback information to the agent. The *resource usage agent* is aware of the hardware characteristics of T2 Plus and knows about the hyper-threading (HT) technology provided by this kind of architecture with up to 8 hardware threads. Therefore the agent suggests to use HT technology to increase FPU utilization and reports to the *design space agent* to explore possibilities to increase the number of threads. Consequently, the *design space agent* proposes to assign the rows of result matrix C to 2, 4, 6 hardware threads per core resulting in a total number of 16, 32, 48 threads, respectively. Three versions are generated and submitted for execution. Moreover, feedback information is used by the compilation system to perform further optimizations (cf. Gupta and Mehofer 2002).

The key point is that this time-consuming tuning task is done automatically by the system and not by the programmer. The different versions are automatically generated and run on T2 Plus, and the monitoring results are reported back to the agents and the programmer. Figure 10.3(b) shows the normalized execution times (longest execution time denoted by time unit 1.0) for the different versions with 1, 2, 4, 6 threads per core and the improvements achieved by the optimizations taking programmer annotations and hardware characteristics into account. The performance improvement of loop interchange is considerable and amounts to 26% for 1 thread per core, approximately 20% for 2 and 4 threads per core, and 66% for 6 threads per core. The performance improvement for increasing the number of threads per core to deal with memory latency is even more significant. The performance improvement assigning 2 and 4 threads to one core was for both loop nest versions approximately a factor of 1.7 and 2.5, respectively. For 6 threads per core, we got for (i, j, k) loop nest a factor of 2.8 and for (i, k, j) loop nest up to 3.7. Based on this experience, the *resource usage agent* classifies increasing the number of threads to deal with memory latency as valuable optimization which has proven beneficial for this processor. The programming environment may suggest this kind of optimization for similar processor architectures as well.

10.5 Related Work

An increasing number of research projects is addressing the challenge of programming multicore computing systems. The *Habanero project* (Habanero Multicore Software Project 2009), which started in Fall 2007 at Rice University, aims to develop languages and compilers for the development of portable software for multicore systems. The *SALSA project* (Service Aggregated Linked Sequential Activities 2009) at Indiana University is investigating the use of services as building blocks for composing parallel data-mining applications based on the workflow paradigm. *Linked Sequential Activities* in SALSA, which are conceptually based on Communicating Sequential Processes of Hoare, are used to build services. The *Berkeley View* (Asanovic et al. 2006) project investigates the influence of multicore processors in applications, hardware, programming models, and systems software for par-

allel computing. The Berkeley View proposes to use a set of *dwarfs* (a dwarf defines a specific computation and communication pattern) for evaluation of parallel programming models. The recently established *Pervasive Parallelism Laboratory (PPL)* (Pervasive Parallelism Laboratory 2009) at Stanford University is investigating future parallel computing platforms. PPL is supported by six computer and chip makers that are convinced that their product sales may decline if software is not able to use effectively the new multicore-based hardware. *SWARM* (Bader et al. 2007), developed at Georgia Institute of Technology, is a parallel programming framework that provides a collection of primitives for programming multicore processors. The *Programming Environments Laboratory (PELAB)* (Programming Environments Laboratory 2009) at Linköping University is investigating the applicability of round-trip engineering techniques to parallelization of sequential programs. The *Cell Superscalar (CellSs)* (Perez et al. 2007) project at Barcelona Supercomputing Center focuses on parallelization of sequential programs for Cell BE processor. The CellSs parallelization involves the functional decomposition, code annotation, and the use of a source-to-source compiler. The *IT Research Division of the NEC Laboratories Europe* (Wagner et al. 2008) is investigating the use of work stealing concept to achieve load balancing. *Performance Portability and Programmability for Heterogeneous Many-core Architectures* (PEPPHER) (PEPPHER 2009) is a related project that is funded under the Seventh Framework Programme of the European Commission. PEPPHER aims at providing a unified framework for programming architecturally diverse, heterogeneous manycore processors to ensure performance portability.

In contrast to the related work, we propose an intelligent programming environment that proactively supports the user during major phases of program development and performance tuning by providing context-specific knowledge and performing iterative time-consuming tasks involved in program development in a semi automatic/autonomic manner.

10.6 Conclusions

We have outlined an intelligent programming environment, which proactively supports the user during high-level program composition, design space exploration, and resource usage optimization. We have highlighted the potential benefits of using such a programming environment with usage-scenarios.

We have observed that even for a rather simple parallel building block such as matrix multiplication, the exploration of the parameter space may be, on one hand, time prohibitive, but, on the other hand, there is a big potential for performance improvement. The example scenario described a first and manageable step toward an intelligent program environment for multicore architectures. Several projects at the authors' home institutions are currently pursued toward the realization of such an intelligent programming environment for multicore computing systems.

Acknowledgements Fatos Xhafa's research work is supported by a grant from the General Secretariat of Universities of the Ministry of Education, Spain. The authors are grateful for numerous discussions and contributions related to Sect. 10.2 to Alastair Donaldson, Hans-Christian Hoppe, Christoph Kessler, David Moloney, Raymond Namyst, Peter Sanders, Jesper Larsson Traff, and Philippas Tsigas.

References

AGEIA Technologies (now a subsidiary of NVIDIA): The PhysX processor, http://www.ageia.com, accessed November 2009.

E. Alba, F. Almeida, M. Blesa, J. Cabeza, C. Cotta, M. Diaz, I. Dorta, J. Gabarro, C. Leon, J. Luna, L. Moreno, C. Pablos, J. Petit, A. Rojas, and F. Xhafa. MALLBA: A library of skeletons for combinatorial optimisation (research note). In Euro-Par 2002. Springer, Berlin, 2002.

M. Alind, M. Eriksson, and C. Kessler. BlockLib: a skeleton library for cell broadband engine. In International Workshop on Multicore Software Engineering (IWMSE-2008) at ICSE-2008. Leipzig, Germany, May 2008. ACM, New York, 2008.

E. Allen, D. Chase, J. Hallett, V. Luchangco, J. Maessen, S. Ryu, G. Steele Jr., and S. Tobin-Hochstadt. The Fortress language specification, version 1.0 (available at http://research.sun.com/projects/plrg/Publications/fortress.1.0.pdf), March 2008.

J. Ansel, C. Chan, Y. Wong, M. Olszewski, A. Edelman, and S. Amarasinghe. PetaBricks: a language and compiler for algorithmic choice. In ACM SIGPLAN Conference on Programming Language Design and Implementation, June 2009.

K. Asanovic, R. Bodik, B. Catanzaro, J. Gebis, P. Husbands, K. Keutzer, D. Patterson, W. Plishker, J. Shalf, S. Williams, and K. Yelick. The landscape of parallel computing research: a view from Berkeley. EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2006-183, December 18, 2006.

AURORA: a priority research program on advanced models, applications and software systems for high performance computing (1997–2007). http://www.vcpc.univie.ac.at/aurora/, accessed November 2009.

D. Bader, V. Kanade, and K. Madduri. SWARM: A parallel programming framework for multicore processors. First Workshop on Multithreaded Architectures and Applications (MTAAP) at IPDPS 2007. Long Beach, CA, USA, March 2007. IEEE, New York, 2007.

S. Benkner. VFC: The Vienna Fortran compiler. Scientific Programming, 7(1):67–81, 1999.

S. Benkner, S. Andel, R. Blasko, P. Brezany, A. Celic, B. Chapman, M. Egg, T. Fahringer, J. Hulman, E. Kelc, E. Mehofer, H. Moritsch, M. Paul, K. Sanjari, V. Sipkova, B. Velkov, B. Wender, and H. Zima. Vienna Fortran compilation system, Version 1.2, user's guide. Technical report, Institute for Software Technology and Parallel Systems, University of Vienna, February 1996.

F. Bodin, S. Bihan. Heterogeneous multicore parallel programming for graphics processing units. Scientific Programming, 17(4):283–348, 2009.

R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon. Parallel Programming in OpenMP. Morgan Kaufmann, San Francisco, 2000.

P. Charles et al.: X10: An object-oriented approach to non-uniform cluster computing. In Proc. ACM OOPSLA'05, Oct. 2005. See also: Report on the Experimental Language X10, Draft 0.41 (available at http://www.research.ibm.com/x10), Feb. 2006.

M. Cole. Bringing skeletons out of the closet: a pragmatic manifesto for skeletal parallel programming. Parallel Computing 30(3):389–406, 2004.

Cray Inc., Seattle, WA. Chapel specification, version 0.780, February 2008. (http://chapel.cs.washington.edu).

A. Donaldson, C. Riley, A. Lokhmotov, and A. Cook. Autoparallelisation of sieve C++ programs. In Proceedings of the 1st Euro-Par Workshop on Highly Parallel Processing on a Chip (HPPC), volume 4854 of *Lecture Notes in Computer Science*, pages 18–27. Springer, Berlin, 2007.

H. Dybdahl, P. Stenström, and L. Natvig. A cache-partitioning aware replacement policy for chip multiprocessors. In 13th Intern. Conf. of High Perform. Comput., HiPC 2006. Springer, Berlin, 2006.

A. Eichenberger et al. Using advanced compiler technology to exploit the performance of the cell broadband engineTM architecture. IBM Systems Journal, 45(1):59–84, 2006.

K. Fatahalian, T. Knight, M. Houston, M. Erez, D. Horn, L. Leem, J. Park, M. Ren, A. Aiken, W. Dally, and P. Hanrahan. Sequoia: programming the memory hierarchy. In Proceedings of the ACM/IEEE SC2006 Conference on High Performance Networking and Computing, November 11–17, 2006, Tampa, FL, USA. ACM, New York, 2006.

I. Foster and C. Kesselman (Editors). The Grid 2: Blueprint for a New Computing Infrastructure. The Elsevier Series in Grid Computing, 2 edition. Morgan Kaufmann, San Mateo, 2003.

R. Gupta, E. Mehofer, and Y. Zhang. Profile guided code optimizations. In Y.N. Srikant, and P. Shankar, editors, The Compiler Design Handbook: Optimizations & Machine Code Generation. CRC Press, Boca Raton, 2002.

Habanero Multicore Software Project. http://www.cs.rice.edu/~vs3/habanero/, accessed November 2009.

M. Hall, Y. Gil, and R. Lucas. Self-configuring applications for heterogeneous systems: program composition and optimization using cognitive techniques. In Proceedings of the IEEE, Special Issue on Cutting-Edge Computing: Using New Commodity Architectures, Volume 96, Issue 5, 2008.

High Performance Fortran Forum. High performance Fortran language specification, version 2.0. Technical report, January 1997.

L. Howes, A. Lokhmotov, A. Donaldson, and P. Kelly. Deriving efficient data movement from decoupled access/execute specifications. In Proceedings of the 4th International Conference on High Performance and Embedded Architectures and Compilers (HiPEAC), volume 5409 of *Lecture Notes in Computer Science*, pages 168–182. Springer, Berlin, 2009.

IBM Cell/B.E. SDK for multicore acceleration version 3.1, available at http://www.ibm.com/developerworks/power/cell/, 2008.

IBM XL C/C++ for Multicore Acceleration for Linux. http://www.alphaworks.ibm.com/tech/cellcompiler, accessed November 2009.

Intel Corp.: Intel Threading Building Blocks 2.1, Reference Manual, 2009 (available at http://www.threadingbuildingblocks.org).

Khronos OpenCL Working Group. The OpenCL specification, version 1.0. Updated, May 16, 2009. http://www.khronos.org/registry/cl/specs/opencl-1.0.43.pdf

T. Knight, J. Park, M. Ren, M. Houston, M. Erez, K. Fatahalian, A. Aiken, W. Dally, and P. Hanrahan. Compilation for explicitly managed memory hierarchies. In Proceedings of the 12th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming, San Jose, California, USA, pages 226–236, ACM Press, New York, 2009.

S. Lindley. Implementing deterministic declarative concurrency using sieves. In Proceedings of the ACM SIGPLAN Workshop on Declarative Aspects of Multicore Programming (DAMP), ACM, New York, 2007.

Model Driven Architecture. http://www.omg.org/mda/, accessed November 2009.

R.W. Numerich and J. Reid. Co-array FORTRAN for parallel programming. SIGPLAN Fortran Forum, 17(2):1–31, 1998.

NVIDIA Corp. CUDA Zone: http://developer.nvidia.com/object/cuda.html, accessed November 2009.

J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. Lefohn, and T. Purcell. A survey of general-purpose computation on graphics hardware. Computer Graphics Forum, 26(1):80–113, 2007.

J. Perez, P. Bellens, R. Badia, and J. Labarta. CellSs: Making it easier to program the cell broadband engine processor. IBM Journal of Research and Development 51(5): 593–604, 2007.

Performance Portability and Programmability for Heterogeneous Many-core Architectures (PEPPHER). The Seventh Framework Programme of the European Commission. Project Reference: 248481. http://www.peppher.eu/, accessed November 2009.

Pervasive Parallelism Laboratory. http://ppl.stanford.edu/wiki/index.php/Pervasive_Parallelism_Laboratory, accessed November 2009.

S. Pllana et al. On customizing the UML for modeling performance-oriented applications. In Proceedings of «UML» 2002, Model Engineering, Concepts and Tools, *LNCS* 2460, Springer, Dresden, 2002.

S. Pllana et al. Teuta: tool support for performance modeling of distributed and parallel applications. In International Conference on Computational Science, Tools for Program Development and Analysis in Computational Science. Krakow, Poland, June 2004. Springer, Dresden, 2004.

S. Pllana, S. Benkner, F. Xhafa, and L. Barolli. Hybrid performance modeling and prediction of large-scale computing systems. In 2008 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008). Barcelona, Spain, March 2008. IEEE CS, Los Alamitos, 2008.

L. Pouchet, C. Bastoul, A. Cohen, and N. Vasilache. Iterative optimization in the polyhedral model: Part I, one-dimensional time. IEEE/ACM Fifth International Symposium on Code Generation and Optimization (CGO'07), San Jose, California, pages 144–156. IEEE Computer Society Press, Los Alamitos, 2007.

Programming Environments Laboratory (PELAB). http://www.ida.liu.se/labs/pelab/, accessed November 2009.

RapidMind corp. home page: http://www.rapidmind.net/, accessed November 2009.

L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, R. Espasa, E. Grochowski, T. Juan, and P. Hanrahan. Larrabee: a many-core x86 architecture for visual computing. ACM Transactions on Graphics 27, 3 (2008), 1–15.

Service Aggregated Linked Sequential Activities (SALSA). http://www.infomall.org/multicore/, accessed November 2009.

D. Shreiner, M. Woo, J. Neider, and T. Davis. OpenGL Programming Guide: The Official Guide to Learning OpenGL. Addison-Wesley Professional, Reading, 2005.

I.J. Taylor, E. Deelman, D.B. Gannon, M. Shields (Editors). Workflows for E-science: Scientific Workflows for Grids. Springer, Berlin, 2006.

The UPC Consortium. UPC Language Specification (v 1.2), June 2005. available at http://upc.gwu.edu.

J. Wagner, A. Jahanpanah, and J. Träff. User-land work stealing schedulers: Towards a standard. 2008 International Workshop on Multi-Core Computing Systems (MuCoCoS'08) at CISIS 2008. Barcelona, Spain, March 2008. IEEE CS, Los Alamitos, 2008.

M. Wooldridge. An Introduction to MultiAgent Systems. Wiley, New York, 2002.

K. Yelick, L. Semenzato, G. Pike, C. Miyamato, B. Liblit, A. Krishnamurthy, P. Hilfinger, S. Graham, D. Gay, P. Colella, and A. Aiken. Titanium: a high-performance Java dialect. Concurrency: Practice and Experience, 10(11–13):825–836, 1998.

Chapter 11
Multimodal and Agent-Based Human–Computer Interaction in Cultural Heritage Applications: an Overview

Antonio Gentile and Salvatore Vitabile

Summary One of the most recent and interesting applications of human–computer interaction technologies is the provision of advanced information services within public places, such as cultural heritage sites or schools and university campuses. In such contexts, concurrent technologies used in smart mobile devices can be used to satisfy the mobility need of users allowing them to access relevant resources in a context-dependent manner. Of course, most of the constraints to be taken into account when designing a pervasive information providing system are given by the actual domain where they are deployed.

This chapter presents an overview of such techniques, focused on two different approaches to the development of human–computer interaction aimed at providing solutions for engaging fruition of cultural heritage sites and exhibits. The chapter will first present multimodality as a key enabler for a more natural interaction with the virtual guide and its surrounding environment. The second approach will be presented next, offering an overview of the evolution of agent-based human–computer interaction systems for the same domain.

11.1 Introduction

The exponential diffusion of small and mobile devices, third-generation wireless communication devices, and location technologies has led to a growing interest toward the development of pervasive and context-aware services. Such technologies,

A. Gentile
Dipartimento di Ingegneria Informatica, University of Palermo, Viale delle Scienze, Ed. 6, 90128 Palermo, Italy
e-mail: gentile@unipa.it

S. Vitabile (✉)
Dipartimento di Biopatologia e Biotecnologie Mediche e Forensi, University of Palermo, Via del Vespro, 90127 Palermo, Italy
e-mail: vitabile@unipa.it

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,* Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_11,

both hardware and software, made the Mark Weiser's vision of Ubiquitous Computing real and more available for users in their in everyday life (Weiser 1991). The Ubiquitous Computing paradigm relies on a framework of smart devices that are thoroughly integrated into common objects and activities. Such a framework implements what is otherwise called a pervasive system, the main goal of which is to provide people with useful services for everyday activities.

As a consequence, the environment in which a pervasive system is operating becomes more complex, that is, enriched by the possibility to access additional information and/or resources on a per-needed basis. Augmented environments can be seen as the composition of two parts: a visible part populated by active users (visitors, operators) and/or objects (inanimate but controlled by some type of artificial intelligence) interacting through digital devices in a real landscape, and an invisible part made of software objects performing specific tasks within an underlying framework. People would perceive the system as a whole entity in which personal mobile devices are used as human–environment adaptable interfaces.

There are many domains where pervasive systems are suitably applied. One of the most recent and interesting applications of pervasive technology is the provision of advanced information services within public places, such as cultural heritage sites or schools and university campuses. In such contexts, concurrent technologies used in smart mobile devices can be used to satisfy the mobility need of users allowing them to access relevant resources in a context-dependent manner. Of course, most of the constraints to be taken into account when designing a pervasive information providing system are given by the actual domain where they are deployed.

In this chapter we will focus on two different ways to approach the development of human–computer interaction aimed at providing solutions for engaging fruition of Cultural Heritage (CH, in short) sites and exhibits. We will next focus on multimodality as a key enabler for a more natural interaction with the virtual guide and its surrounding environment. In this first part, we also evaluate the approaches examined in the chapter in terms of multimodality, presence of pervasive access to contents, and intelligent handling of interaction with the user. We will then offer an overview of the evolution of agent-based human–computer interaction, reviewing some of the most relevant papers in the past years.

11.2 Multimodal Human–Computer Interaction in Cultural Heritage Applications

Cultural Heritage applications pose tremendous challenges to designers under different aspects. Firstly, because of the large variety of visitors they have to deal with, each with specific needs and expectations about the visit. Secondly, no two sites are the same, and pretty much you need a framework that can easily produce a new installation given the site characteristics (indoor versus outdoor, distributed versus centralized, individual centered versus group centered, etc.). Lastly, the technologies involved must be robust to failures, redundant, and, above all, easy and intuitive to

use. These are the reasons why there are several research groups that are focusing their attention on this applicative domain. It is a good test bed to validate almost all models and design choices.

Multimodality is usually described of such software applications or computing systems that combine multiple modalities of input and output. Being free to choose among multiple modes to interact with such augmented environments and systems is crucial to their wider acceptance by a vast variety of users. This is often the case when such systems are intended for mass fruition at museums or cultural heritage sites. Multimodality is the capability of a system to allow for multiple modes of interaction, from traditional point-and-click to voice navigation or gesture activation of controls. A pervasive system targeted to cultural heritage fruition, therefore, can no longer be designed without first addressing how it will handle interaction with users. Additionally, multimodality relies on redundant information, resulting on more dependable systems that can adapt to the needs of large and diverse groups of users, under many usage contexts.

11.3 A Timeline of Cultural Heritage Fruition Applications

Several workgroups focused their research in the past five years on the definition of models aimed at providing users of a cultural heritage site with useful services of different kinds. In this section, we will introduce some of the projects that we consider most relevant.

Specifically, we will focus on technologies used to access devices, positioning/location, human–environment interfaces, and ambient intelligence. In order to support our review, we will discuss a selection of that we deem best representing the evolution of systems for fruition of cultural heritage sites. Figure 11.1 depicts the timeline we will follow, along with the projects discussed in this chapter.

The projects presented during 2005–2006 witness the need to combine more than one technology at a time, concurrently used to implement multichannel interaction. As a matter of fact, during this timeframe, we assist to the spreading of several technologies (Wi-Fi, Bluetooth, RFID, GPS, voice recognition and synthesis, image recognition, conversational agents) and smart access devices (such as PDA and smartphones) that more and more are integrated into enhanced systems.

Farella et al. (2005) presented a work that exploits widely available personal mobile devices (PDAs and cellular phones) and software environments (usually based

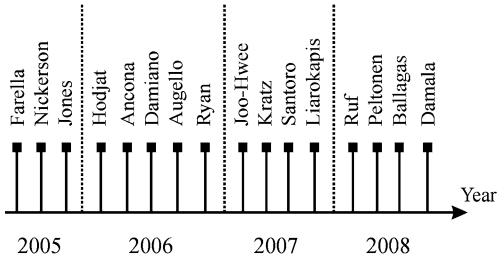


Fig. 11.1 Timeline

on Java) to create highly interactive Virtual Heritage applications based on popular, low-cost, wireless terminals with highly optimized interfaces. Ancona et al. (2006) proposed a system called AGAMEMNON that uses mobile phones equipped with embedded cameras to enhance the effectiveness of tours of both archaeological sites and museums.

The evolution of speech synthesis techniques allow for a new multimedia content delivery way, based on text-to-speech technology. This allows providing the user with new services accessible also by telephone. This solution tries to overcome the well-known limits of static prerecorded audio guides, by vocally presenting a text that can be dynamically composed. Using this technology, Nickerson (2005) proposes the system named History Calls. He developed an experimental system using VoiceXML technology that is capable to deliver an automated audio museum tour directly to cell phones.

The main target of some research group is to make the visit within a cultural heritage site as more natural as possible for the largest part of users, avoiding the use of ad hoc or complex devices. This goal can be achieved by integrating different well-known technologies in a different way.

Augello et al. (2006) proposed a multimodal approach for virtual guides in cultural heritage sites that enables more natural interaction modes using off-the-shelf devices. The system, termed MAGA, integrates intelligent conversational agents (based on chatbots) and speech recognition/synthesis in an RFID-based location framework with Wi-Fi-based data exchange. Moving along the line of a more natural interaction, Jones et al. (2005) proposed a system in which multiple virtual guides interact with visitors, each characterized by its own personality. In their system, it is then possible to interact with two intelligent agents, with divergent personalities, in an augmented reality environment. Another example of system that is capable to adapt itself to the user behavior is CRUSE (Context Reactive User Experience) proposed by Hodjat et al. (2006). It is a user interface framework that enables the delivery of applications and services to mobile users.

In Dramatour (Damiano et al. 2006), Carletto, a virtual guide for the Savoy apartments of Palazzo Chiablese (in piazza Castello—Torino) provides users with information on portable devices presenting them in a dramatized form (being himself a small spider, long-time inhabitant of Palazzo Chiablese).

The past two years (2007–2008) see the advent of ambient integrated guides that are based on mixed reality, where real physical elements interact with virtual ones. Liarokapis and Newman (2007) focus their work on the design issues of high-level user-centered Mixed Reality (MR) interfaces. They propose a framework of a tangible MR interface that contains Augmented Reality, Virtual Reality, and Cyber Reality rendering modes. This framework can be used to design effective MR environments where participants can dynamically switch among the available rendering modes to achieve the best possible visualization. Santoro et al. (2007) propose a multimodal museum guide that provides user with gesture interaction mode: using a PDA equipped with a 2D accelerometer, they control interface and content navigation by means of hand tilt gestures.

Damala et al. (2008) present a prototype of an augmented reality mobile multimedia museum guide, also addressing the full development cycle of the guide, from

conception to implementation, testing, and assessment of the final system. They use last-generation ultra-mobile PCs, which allow designers to exploit some of the most powerful software technologies, such as OpenCV for video acquisition, ARToolkitPlus for tracking the paintings, OGRE3D for the insertion of the virtual objects, Open AL for the audio output, and XERCES for XML document parsing.

Overall, in the past five years the attention of researchers in the field of pervasive service provision moved to a higher level of abstraction, mainly a middleware level. Software technologies have been exploited to give systems more adaptability, usability, and, above all, intelligence. These features are the more promising ones in order to achieve the naturalness needed to get the largest acceptance of virtual guide systems among common people. As a consequence, the current research interests in the field of service provision in cultural heritage sites are mainly aimed at reaching these features. A common line researchers are following is the concurrent use of multiple hardware and software technologies at a time. The goal is to give users the possibility to interact in multiple modes, according to their habits, skills, and capabilities. This will have the desired side effect to expand the number of prospective users, allowing also disabled ones to exploit such guide systems with their residual capabilities.

11.4 Multimodal Mobile Access to Services and Contents in Cultural Heritage Sites

The cultural heritage target domain is extremely challenging for the development of systems that assist users during their visit according to their mobility requirements. Moreover these systems should attract and involve users by showing an easy and friendly access. Most of these requirements are fulfilled by means of personal mobile devices, suitably applied as adaptive human–system interfaces.

We start our discussion by summarizing the main features of those of the projects previously discussed that make use of hand-held devices (Table 11.1). Systems are listed by year of presentation, along with their features listed under four main categories: compass/position detection, context awareness, intelligent interaction, and output and input modes.

The *compass/position detection* column shows technologies used to detect the user position and orientation within the cultural heritage site. All positioning-based systems exploit information about the proximity of the user to a particular item or point of interest. With no further information, such systems are not able to detect or estimate what the user is looking at, thus making this solution unsuitable for sites with items that are close one to each other. To disambiguate such situations, information about the spatial orientation of the user within the environment is needed in addition to his position. To this end, some systems use different combinations of technologies and techniques, such as infrared combined with electronic compass or with accelerometers embedded into the hand-held devices, thus providing users with fine-tuned contents. Simpler, cheaper, and less intrusive location frameworks

Table 11.1 System features

Year	System	Compass/ position detection	Context awareness	Intelligent interaction	Input modes	Output modes
2005	Farella	–	–	–	keyboard, touch pen or touch-screen,	text, prerecorded audio, video clip, images, virtual reality
2005	Nickerson	–	–	–	vocal interface	prerecorded audio, synthesized narrations
2005	Jones	GPS	location-based, state-based, profile-based	natural language, proactivity, character	keyboard, touch pen or touch-screen, user position and/or compass	text, prerecorded audio, images
2006	Hodjat (CRUSE)	GPS	location-based, state-based, profile-based	natural language, proactivity	keyboard, touch pen or touch-screen, user position and/or compass	text
2006	Ancona		state-based, profile-based	proactivity	keyboard, touch pen or touch-screen, vocal interface, image recognition	text, prerecorded audio, synthesized narrations, video clip, images
2006	Damiano (Drama-Tour)	–	state-based	proactivity, character	keyboard, touch pen or touch-screen, user position and/or compass	video clip

are made available by means of the RFID technology. In fact, the short tag detection distance can be used to estimate the user interest for a specific item with a good accuracy.

Table 11.1 (Continued)

Year	System	Compass/ position detection	Context awareness	Intelligent interaction	Input modes	Output modes
2006	Augello (MAGA)	RFID	location-based, state-based	natural language, inference, proactivity, character	keyboard, touch pen or touch-screen, vocal interface, user position and/or compass	text, synthesized narrations, images
2007	Santoro	RFID	location-based, state-based	proactivity	keyboard, touch pen or touch-screen, gesture caption, user position and/or compass	text, pre-recorded audio, video clip, images
2007	Joo-Hwee	–	–	–	image Recognition	text, audio, images
2008	Ruf	–	location-based	–	image Recognition	text, audio, images, virtual reality

The *Context-awareness* column reports the different techniques used to build context-related contents. Systems that miss this feature provide users with information that is neither position- nor profile-based. Three different methods have been identified for contents composition to make the human–system interaction more natural and interesting:

- *location-based*: users are provided with information related to items that are located near their current position;
- *profile-based*: delivered contents are generated according to the user's profile, such as preferences, skills, age, etc.;
- *state-based*: information are generated taking into account different context factors, such as the user position, the user's profile, the interaction flow, the followed path, etc. Systems may use a subset of these elements to detect the current state (e.g., the history of inputs).

Research results in the field of artificial intelligence suggest new tools to make the interaction more natural (Jones et al. 2005; Ibanez et al. 2003; Almeida and

Yokoi 2003). To take this trend into account, in the *Intelligent Interaction* column we then report the capability of a system to implement methodologies that are typical in the field of artificial intelligence. Specifically, the *proactivity* feature shows that the system can spontaneously initiate the interaction with the user without his explicit request, according to the detected context. Systems that support *natural language* interaction accept user input in natural language (e.g.: "give me information about marble statues dating from the 5th century BC") and/or provide user with spoken information. Systems with the *character* feature embed a life-like tour virtual assistant with a specific personality. *Inference* means that a system has the capability to make inferences on domain ontology to update its knowledge base and to generate ad hoc contents.

The *Input Mode* column lists the input modes available for each system, whereas the *Output Modes* column lists the content delivery modes a system is enabled to use.

In the following we will discuss how issues and problems of multiple concurrent interaction modes have been faced in the systems and approaches examined so far. In particular we now focus our attention on concurrent input modes, as most of researchers we cited did. Actually, the processing of multiple inputs simultaneously coming from different channels present several constraints, for instance, in terms of synchronization and concurrency, whereas the management of contemporary output modes is less compelling. To this end, in Fig. 11.2 we depict the temporal evolution of such systems, comparing them under three dimensions: degree of multimodality, pervasive access to contents, and intelligence in interaction management.

Any given system is therefore classified as either *none, intelligent*, *pervasive*, or *both* according to the performed interaction, as illustrated in the figure with the bar

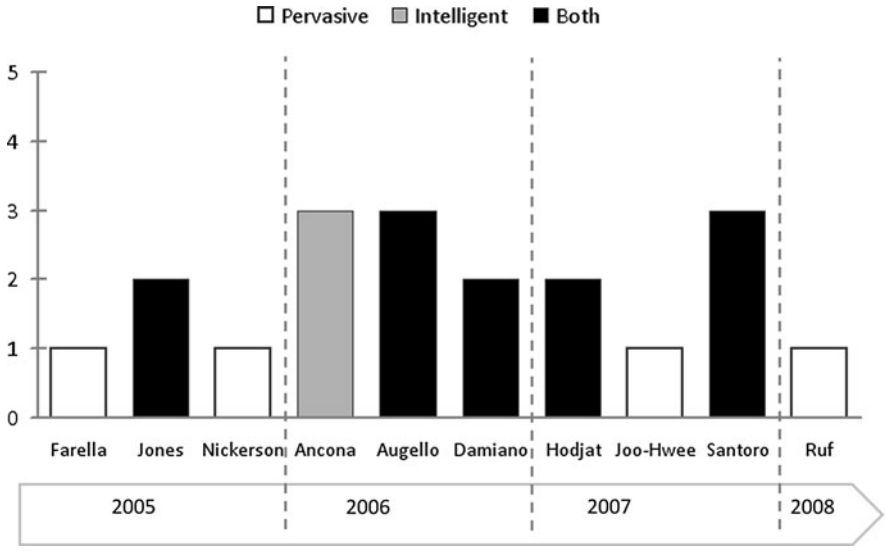


Fig. 11.2 Evolution of multimodal input and performed interaction

filling pattern. Systems classified as *none* do support neither intelligent interaction nor pervasive access. A system offers *pervasive* access to cultural heritage contents if it relies on a framework of smart devices where data are stored and processed. The retrieved information is then suitably formatted to fit the specific access device and transmitted over a wireless connection. We classify as *intelligent* in the interaction a system in which contents generation and/or delivery are managed with techniques that are inherited from the artificial intelligence research field. In particular, from the contents generation point of view, the system should use one of the following methods: state-based, location-based, profile-based, and inference. From the contents delivery point of view, the system should either interact using natural language, or exhibit proactive behavior, or show a life-like character, or any combination of the three.

The degree of *multimodality* of each system is presented on the vertical axis, and it is evaluated by counting the number of input modes by which users gain access to information.

Figure 11.2 shows that in the last decade the pervasive approach in service provision within cultural heritage sites has been largely adopted. This is mainly due to the distributed nature of data and to the need of accessing those data anywhere at any time. The figure also shows that the use of algorithms and techniques of the Artificial Intelligence field is considered a key feature for the success of a system, particularly when the main goal is the naturalness of the interaction. Pervasive systems accessed by mobile devices substituted the prerecorded audio guides, providing data mining algorithms with higher computational power and with more data sources.

In order to better highlight the evolution of interaction, in Table 11.2 we present the detailed list of input modes implemented by each of the discussed systems. This allows readers to realize that, despite new interaction modes are studied and made available, most of existing systems keep showing the traditional point-and-click interface, although being implemented by up-to-date devices.

Table 11.2 Input modes

Year	System	Keyboard, touch pen or touch-screen	Vocal interface	User position and/or compass	Gesture recognition	Image recognition
2005	Farella	X				
2005	Nickerson		X			
2005	Jones	X		X		
2006	Hodjat	X		X		
2006	Ancona	X	X			X
2006	Damiano	X		X		
2006	Augello	X	X	X		
2007	Santoro	X		X	X	
2007	Joo-Hwee					X
2008	Ruf					X

Another commonly used input mode is the user position. In fact, this allows designers to give their systems the context-awareness by taking into account an important context element. Furthermore, the needed framework can be relatively cheap and often may partly rely on some existing implementation (e.g., the GPS).

Different input modes, such as voice, gesture, and image recognition, are commonly considered as useful in order to improve the naturalness of interaction, but they have been not so largely adopted mainly due to their complexity. Advances of both hardware and software technologies in these fields are giving an important stimulus to their adoption, and several research groups are currently working in the field of integration of these technologies.

11.5 Agent-Based Human–Computer Interaction in Cultural Heritage Applications

There is a very large number of research projects coping with the problem of providing information about sites of cultural heritage value (see Table 11.3 for a list of the most relevant research projects). Complex distributed applications in very challenging domains, such as e-Business, e-Government, Cultural Heritage, etc. need to use new models to overcome classical client–server and web-based model limits. In a cultural heritage site, different mobile or fixed devices interact with a great number of services. As example, the number of devices in the museum cannot be easily predicted as well as the number of the available and active services. Environment is highly dynamic: users enter and leave the site, and availability of services cannot be guaranteed (service providers may be busy or simply off-line).

Accordingly to Ducatel et al. (2001) and Penserini et al. (2003), paradigms in which software applications are constructed based on independent component services with interfaces can be seen as a feasible solution (Service-Oriented Computing and Service-Oriented Architectures) for CH applications. Intelligent Agents are a natural choice to implement and develop the above paradigm. In fact, each service can be seen as one or more interacting autonomous agents. At the individual-agent level, each agent requires a representation of the individual behavior through elements such as beliefs, response, intentions, desires, goals, and ego. In addition, each agent-based service should provide intelligent functionalities, such as reactivity, proactivity, and social ability. On the other hand, those applications need intelligent mechanism to exhibit, compose, and adapt its behaviors to external changes and/or triggers.

Accordingly to Lopez-Jaquero et al. (2009), Agent-Based Human–Computer Interaction (AB-HCI) supports the execution of human-centerd applications by means of one or more interface agents aimed at providing an advanced user interaction experience by processing the incoming information from the environment and applying techniques coming from different disciplines (human–computer interaction, software engineering, artificial intelligence, psychology, . . .). In the cultural heritage domain, the required software architecture can really benefit from using intelligent

Table 11.3 System features

Project	System	Compass/ position detection	Context awareness	Intelligent interaction	Input modes	Output modes
Minerva	Amigoni and Schiaffonati (2003, 2009)	–	–	natural language	keyboard, natural language	HTML pages, VRML objects
Teschet	Pilato et al. (2004a)	–	–	inference	keyboard, touch pen or touch-screen	Text
DramaTour	Damiano et al. (2006)	–	state-based	proactivity, character	keyboard, touch pen or touch-screen, user position and/or compass	video clip
Row 7 MAGA	Augello et al. (2006)	RFID	location-based, state-based	natural language, inference, proactivity, character	keyboard, touch pen or touch-screen, vocal interface, user position and/or compass	Text, synthesized narrations, Images
PEACH	Stock et al. (2007)	IR, IRDA, RFID	location-based, profile-based	proactivity	keyboard, touch pen or touch-screen	video clip
MOSAICA	Shah et al. (2007)	–	–	–	keyboard, mouse, touch-screen	images, textual and vocal description
Cuspis	Costantini et al. (2008)	satellite signals	location-based, profile-based	inference, proactivity	user position	Text

agents, since they support a much more natural method to design decision-making mechanisms by providing constructs closer to the ones used in human reasoning theories.

At the same time, agent-based systems offer different key benefits. First of all, MAS is the natural paradigm allowing for the distribution of computation, making

Table 11.3 (Continued)

Project	System	Compass/ position detection	Context awareness	Intelligent interaction	Input modes	Output modes
–	Lopez-Jaquero et al. (2009)	–	environment-based, profile-based, device-based	proactivity, adaptivity	keyboard, touch pen or touch-screen	text, images, vocal
AB-MCUI	Huang et al. (2009)	camera, sensor, accelerator	cultural-based, state-based	speech recognition, non-verbal behavior, character	vocal interface, user position and/or compass, user gesture and movements, cultural information	recorded audio, animations

easier the integration with current trends in software design such as service-oriented application development. The main idea is the efficient utilization of the resources available to a mobile device in its vicinity, whether it is in the wired network or whether the mobile device is in an ad hoc network. Accordingly to Lind (2001), a MAS consists of multiple independent entities, which coordinate with themselves in order to achieve their individual and their joint goals. In addition, agent-based design methodologies can be extended to the programming of massively parallel computing platforms, since an agent can be seen as the instance of an agent class that is the software implementation of an autonomous entity capable of pursuing an objective through its autonomous decisions, actions, and social relationships (Gentile et al. 2002). In designing a mobile-based application, certain restrictions are applied, such as limited memory and processing resources. Each agent will be represented to solve a scheduled problem. In this view, the mobile device service application handles the mobile user interface that relates to the interaction between the application and the user, covering user's data-entry and portable device output functions (Abdel-Naby et al. 2007).

On the other hand, a user-centered approach adapted to the design of collaborative technologies is more accurately referred to as a group-centered approach (McNeese et al. 1995). In order to support a user-centered approach to the design of collaborative technologies, it becomes necessary to acquire knowledge about the group, the group work domain, and the group design requirements directly from the group. In a MAS, separate software entities are forming Virtual Communities

(VC) (Rakotonirainy et al. 2000). Software entities of each VC collaborate to obtain certain results, and they usually share the same interests and goals. The most usual features of CSCW (Computer Supported Cooperative Work) applications are commitment (goals), cooperation, coordination, and competition (Kling 1991). So intelligent agents supporting group behaviors and goals can be seen as a powerful tool to develop GDSS (Group Decision Support Systems) for group-centered applications.

In literature, two main approaches have been followed to design and implement AB-HCIs. The first approach exploits Multi-Agent Systems (MASs) facilities and capabilities for designing and developing cultural heritage solutions and applications. A generic MAS gives the freedom to develop specialized agents for intelligent reasoning tasks and for the interaction between human and computer. Agent communication and interaction are implemented exploiting the low-level functionalities supplied by existing frameworks and platforms for agent development. The second approach deals with the development of relational and conversational agents designed to create and maintain long-term social and emotional relationships with users. In the following, the most meaningful solutions for the two approaches are briefly described.

11.5.1 Multi-Agent System-Based Solutions for AB-HCI

In Amigoni and Schiaffonati (2003) a multiagent system, called Minerva, supporting the creative work (the preparation and the allocation of a virtual museum or a virtual art exhibition) of museum organizers is presented. Minerva is related to archaeological findings and provides, through a graphical interface, different level (role-based) user interaction. From the architectural point of view, Minerva is composed of components and agents. The user interface component generates HTML pages with VRML objects, as response to a user request. The user, through a graphical interface with drop down menus and free text fields, chooses some parameters and sends the request to the Minerva system. A natural language processing component implements basic natural language understanding techniques for a subset of Italian language. Minerva architecture is completed by two main intelligent agents interacting with the above components: the Preparator Agent determines, on the basis of the user selected criteria, the rules to display the works of art of a collection. The Allocator Agent is able to find, within a selected environment, the right position for the works of art in the museum rooms taking into account some constraints (the number of works, room order and position, etc.). In Amigoni and Schiaffonati (2009) an evolution of the previous architecture is presented. The new architecture is a MAS-based architecture in which multiple clients interact with the Minerva server and, consequently, with the Preparator Agent and Allocator Agents (one for each client). VRML builder agents build the requested views. The new architecture has been developed using the JADE platform.

In Pilato et al. (2004a, 2004b) a Multi-Agent System (MAS) for automatic and concurrent document retrieval in the cultural heritage domain was presented. The

system was composed by four agents: a Trainer Agent, a mobile Neural Classifier Agent, a Librarian Agent, and finally an Interface Agent. The system was based on both the mobile agent paradigm and neural network architecture for a sub-symbolic knowledge representation. The main feature of the system was its versatility in managing documents whose topic class is unknown and not a priori fixed: the system autonomously adapts its document classification capability, exploiting the web directories available in the most common search engines. The entire system was developed using the JADE (Java Agent DEvelopment Framework) platform (Bellifemine et al. 1999). Concerning user interaction, the Interface Agent (IA) implemented a simple Graphic User Interface with drop down menus and provides, on Personal Digital Assistant devices, a front end application to the end user, for checking user inputs and displaying results. The work was developed within the TESCHET (A Technology System for Cultural Heritage in Tourism) research project, having the goal to create a multichannel platform based on pervasive, intelligent, and agent technologies with ontological classification of information for the Italian tourism and cultural heritage domains.

In Stock et al. (2007) the experience and the results of the PEACH (Personal Experience with Active Cultural Heritage) project are reported and analyzed. The PEACH project, located in the Castle of Buonconsiglio in Trento (Italy), was aimed to create an interactive and personalized guide for enhancing cultural heritage enjoyment through individual's background, needs, and interests profiling. From our perspective, the system is composed of two main (interdependent) components: a three-tier "classical" application containing a presentation layer (User Assistant—UA) and a MAS to provide the required services and interactions. Agents communicate in order to provide relevant and personalized presentations to the museum visitor based on his/her location and interest. The UA runs on the user's PDA and provides the system interface, while a Presentation Composer receives explicit and implicit user requests (user interest propagation) and replies with appropriate presentations (small Flash presentations). PEACH focus is essentially on presentations personalization, incorporating the information supplied from a positioning system and the concept of situation-aware content. Peach also focus on the concept of Active Museum, an intelligent and pervasive environment.

In Shah et al. (2007) an agent-based approach to develop the Semantically Enhanced, Multifaceted, Collaborative Access to Cultural Heritage (MOSAICA) pedagogical framework is proposed. MOSAICA is organized as an advanced web portal and has the purpose to design a toolbox for intelligent presentation, knowledge-based discovery, and interactive and creative interactions covering a broad variety of cultural heritage resources. The initial focus of MOSAICA was Jewish cultural heritage. The MAS-based approach (the system is composed of two interacting agents) is used to develop virtual expeditions as specific educational instruments based on conceptual modeling and designed for learning through exploration of virtual worlds. MOSAICA's framework is composed of several navigational interfaces, integrating documents, images, and GIS data for virtual explorations.

In Costantini et al. (2008) the DaliCa multiagent system, exploiting intelligent agents, was proposed. DaliCa was developed as central component of the European

Cultural Heritage Space Identification System (Cuspis) project, and it addresses the dissemination of information about cultural assets. DaliCa was successfully tested at the University of L'Aquila and at the Villa Adriana (Rome, Italy) area. The designed MAS application consists of three application agents and three application environment components. When a new user starts on a visit, the generator agent produces an initial user profile agent that is able to monitoring visitors' interests and behavior. An output agent performs information exchange between DaliCa and external infrastructures. The application environment has three components as well. The ontology interface gives agents information about the cultural heritage site context. The visitor interface sends to the DaliCa systems visitors' positions and data (the Galileo satellite signal is used for the visiting Point Of Interest localization). When a new visit starts, the user profile agent elaborates either the data coming from the initial profile or the new data derived from the user behavior, deducing visitor interests. Successively, the agent suggests the most appropriate sites to the visitors through their PDAs. DaliCa was implemented in DALI, a logical-agent-oriented language (Costantini and Tocchio 2004).

In Lopez-Jaquero et al. (2009) an adaptive interface based on a set of collaborative agents to assist the user in handling some tasks in a museum is proposed. The architecture is able to detect the context of use (AgentDetectContextOfUse) through the received information about the characteristic of the adopted platform and devices (AgentContextPlatform), about the user's goals (AgentContextUser), and about the characteristics and the changes of the environment (AgentContextEnvironment). The information about the perceived context of use is forwarded to the Agent Adaptation Process for the selection of the interaction rules fitting the supplied information. Moreover, agents update the context-of-use models (user, platform, and environment) to keep an up-to-date view of the context where the interaction takes place. As example, in the museum, the orientation of the screen in the PDA could be changed by the user. This change is automatically detected by means of software sensors (AgentDetectContextOfUse) and forwarded to the Agent Adaptation Process for the activation of the related adaptivity rules.

11.5.2 Conversational Agent Based Solutions for AB-HCI

Embodied Conversational Agents (ECA) are computer-generated human-like characters that improve the naturalness of the interaction between humans and computers. To achieve that feature, agents assemble several intelligent features, such as natural language understanding and generation, sensor data processing, gesture recognition and generation, personality modeling, facial expression recognition and generation, and so on. In the previous sections, two examples of conversational agents have been presented and described.

In Augello et al. (2006) a conversational agent for achieving natural interactions and enabling site fruition also by inexpert and/or disabled users was proposed. The conversational agent is enriched by a speech recognition/synthesis module and an

RFID-based auto-localization module. The agent integrates also reasoning capabilities, since an ontology for the specific domain was firstly created and then combined with the agent dialogue module.

In Damiano et al. (2006) a character, representing a teenage spider with an anthropomorphic aspect (Carletto), has been designed to create an interactive guided tour. The application was tested in the historical site of Palazzo Chiablese in Turin (Italy): Carletto's family has inhabited the palace from ages, so that the spider knows either the history of the palace or a lot of funny anecdotes. Carletto has been designed and developed through a new methodology, called DramaTour, for creating information presentations based on a dramatization. The methodology has a modular structure integrating the handling of user interactions, the content organization, and the final delivery of audiovisual contents.

In Huang et al. (2009) a general purpose framework to build an ECA-based customer application is proposed. The framework was developed as part of the "An Agent Based Multicultural User Interface in a Customer Service Application" (ABMCUI, in short) project. A characterized tour guide of Dubrovnik, Croatia, answers queries coming from human visitors with verbal and nonverbal interactions. The tour guide is able to recognize and interact with Japanese or Croatian visitors, adapting itself to the Japanese mode (speaking and behaving following Japanese rules) or to the Croatians mode (speaking and behaving following European rules). User interaction is implemented through natural language speaking and nonverbal behaviors such as pointing to an object on the background image. Advanced modules for head orientation tracking, hand shape recognition, and head nodding/shaking recognition have been also implemented. Tour guide design and development have been performed using the GECA (Generic ECA) framework. GECA is composed of a low-level communication platform, a set of communication API libraries, and a high-level protocol (XML-based messages).

11.5.3 Discussions and Comparisons

In Table 11.3, there are summarized the main features of the previously discussed projects that make use of hand-held devices, PDAs, or laptop. Systems are listed by year of presentation, along with their features listed under the same four main categories previously listed: compass/position detection, context awareness, intelligent interaction, and output and input modes.

As stated before, the *compass/position detection* column shows technologies used to detect the user position and orientation within the cultural heritage site. The most recent solutions use location systems to detect the proximity of the user to a particular item or point of interest. Some systems use also advanced techniques (IR, RFID) to obtain information about the spatial orientation of the user within the environment. In Huang et al. (2009) an interesting approach based on data coming from IR and CCD cameras, motion, tracking, and accelerator sensors is proposed.

As previously defined, the *Context-awareness* column reports the different techniques used to build context-related contents. In our analysis three new methods

have been identified for contents composition in HCI (see Sect. 11.4 for *location-based*, *profile-based*, and *state-based* definitions):

- *environment-based*: the system is provided with information about the physical environment (for example, light condition);
- *device-based*: the system is provided with technical information about the used mobile device (display features and resolution, device orientation, etc.);
- *cultural-based*: the system is provided with information about visitor origin and culture in order to address human–computer interaction with typical movements, speeches, and expressions.

As stated before, the *Intelligent Interaction* column reports the capability of a system to implement methodologies that are typical in the field of Artificial Intelligence or Computational Intelligence. For this purpose, we have also added *speech recognition* and *nonverbal behavior* (gesture, motion, and expression recognition) among the items for intelligent interaction.

The *Input Modes* column and the *Output Modes* column report the input modes available for each system and the content delivery modes a system is enabled to use, respectively.

In Table 11.4, the most meaningful system implementation features are summarized. In the most of cases, MAS-based HCI has been developed using an agent development methodology, an agent platform, or both. Agent development methodologies are software tools for designing and developing multiagent societies. They often integrate design models and concepts from both Software Engineering and Artificial Intelligence approaches. Among the agent development methodologies used in the cultural heritage domain, there are the PASSI (a Process for Agent Societies Specification and Implementation) methodology (Cossentino and Potts 2002), the Tropos methodology (Giunchiglia et al. 2002), and the Prometheus methodology (Padgham and Winikoff 2005). JADE (Java Agent DEvelopment Framework) platform is a software framework that facilitates the implementation of multiagent systems through a middleware and a set of graphical tools for the debugging and deployment phases (Bellifemine et al. 1999). Analogous facilities offer the GECA framework used in Huang et al. (2009). DaliCa multiagent system (Costantini et al. 2008) was developed and implemented using the logical-agent-oriented DALI language (Costantini and Tocchio 2004). So in cultural heritage domain, agent design and development tools and frameworks to simplify application development are widely used for MAS-based HCI . However, they do not implement a character-based interaction, even if it is the most emotional and natural interface for HCI.

In contrast with MAS-based HCI, single-agent-based human–computer interaction is implemented embedding a life-like virtual character with a specific personality. Generally, no methodology or platform are used to develop conversational agents, even if Dramatour (Damiano et al. 2006) has been designed and used for the anthropomorphic spider design.

Table 11.4 System implementation details

System	MAS based	Character based interaction	Agent development methodology	Agent platform/ language
Pilato et al. (2004a)	YES	NO	PASSI	JADE
Damiano et al. (2006)	NO	YES	Dramatour	–
Augello et al. (2006)	NO	YES	–	–
Stock et al. (2007)	YES	NO	TROPOS	–
Shah et al. (2007)	YES	NO	–	–
Costantini et al. (2008)	YES	NO	–	DALI
Lopez-Jaquero et al. (2009)	YES	NO	Prometheus	–
Huang et al. (2009)	NO	YES	–	GECA
Amigoni and Schiaffonati (2009)	YES	NO	–	JADE

11.6 Conclusions

Cultural heritage fruition and communication are an exciting new arena to exercise many enabling technologies and study novel, more natural interaction schema, all aimed at engaging visitors with multisensorial, memorable visit experiences. We have thus explored systems for cultural heritage fruition looking at their capability to engage visitors with pervasive, multimodal, intelligent access to information contents. We have particularly focused on multimodality as it is a key enabler for natural, unobtrusive interaction with exhibits and site virtual environment. In addition, we have also looked at how pervasive access to information contents is deployed and to what degree agent-based designs may provide some intelligence to resemble human tracts in system responses to visitors' queries and preferences.

It appears that, in recent years, applications for cultural heritage fruition have focused on locating visitor position inside the environment, as this piece of information is a key to contextualize dynamic contents and offer a natural interaction, specifically geared for the exhibit/artifact at hand. Voice-based techniques are used to a lesser extent, due to the difficult tuning that vocal interface often require. The development of robust speech and speaker recognition systems that operate reliably in crowded, noisy environments is still an open research. An interaction mode that is receiving a growing interest is movement and gesture recognition. In the literature, many exemplary projects are available that propose immersive fruition of virtual world and augmented reality, in which gesture-based commands are intuitive for the visitor.

On the other hand, agent-based HCI provides intelligent functionalities, such as reactivity, proactivity, and social ability. MAS paradigm gives the freedom to develop specialized intelligent agents for high-level reasoning tasks, while ECAs allow one to develop natural and emotional interaction between humans and humanized computers. Several frameworks for MAS development have been developed, so that MAS-based HCI can exploit the low-level functionalities supplied by exist-

ing frameworks and platforms for agent development, communication, and interaction.

Every year, eighty percent of the visitors of cultural heritage sites and museums keep a cellphone in their pockets. It is estimated that in excess of fifty percent of those devices are Java enabled and capable to access Bluetooth or WiFi networks. Whichever their language, they might be tapping into a local server to access their site guide, maybe previously customized on the web before heading off their visit. Those visitors may enjoy a different visit experience, where multimodal interaction with the site, mediated by their own personal device, will result in lasting memories to treasure and relive, once back at home, with family and friends. Enabling this vision could be the mission for Human–Computer Interaction studies on next generation cultural heritage applications.

References

Abdel-Naby, S.; Giorgini, P.; Weiss, M. Design patterns for multiagent systems to elevate pocket device applications. *Agents World (ESAW'07)*. NCSR "Demokritos", Athens, Greece, 22–24 October 2007.

Almeida, P.; Yokoi, S. (2003). Interactive character as a virtual tour guide to an online museum exhibition. *Proceedings of Museum and the Web*.

Amigoni, F.; Schiaffonati, V. (2003). The Minerva multiagent system for supporting creativity in museums organization. *Proceedings of the IJCAI2003 (Eighteenth International Joint Conference on Artificial Intelligence), Workshop on Creative Systems: Approaches to Creativity in AI and Cognitive Science (IJCAI)*, Acapulco, Mexico, 9–10 August 2003, pp. 65–74.

Amigoni, F.; Schiaffonati, V. (2009). The Minerva system: a step toward automatically created virtual museums. *Applied Artificial Intelligence*, 23(3), pp. 204–232.

Ancona, M., Cappello, M., Casamassima, M., Cazzola, W., Conte, D., Pittore, M., Quercini, G., Scagliola, N., Villa, M. (2006). Mobile vision and cultural heritage: the AGAMEMNON project. *Proceedings of the 1st International Workshop on Mobile Vision*, Austria, May 2006.

Augello, A., Santangelo, A., Sorce, S., Pilato, G., Gentile, A., Genco, A., Gaglio, S. (2006). MAGA: a mobile archaeological guide at Agrigento. *Proceedings of the Workshop "Giornata Nazionale su Guide Mobili Virtuali 2006" (ACM-SIGCHI)*, Italy, Torino, 18 ottobre 2006, online proceedings http://hcilab.uniud.it/sigchi/doc/Virtuality06/index.html.

Bellifemine, F., Poggi, A., Rimassa, G. (1999). JADE: a FIPA-compliant agent frame work. *CSELT internal technical report. Part of this report has been also published in Proceedings of PAAM'99*, London, April 1999, pp. 97–108.

Cossentino M., Potts C. (2002). A CASE tool supported methodology for the design of multi-agent systems. *Proceedings of the 2002 International Conference on Software Engineering Research and Practice (SERP'02)*, Las Vegas, NV, USA, 24–27 June 2002.

Costantini, S., Tocchio A., (2004). The DALI logic programming agent-oriented language. *Logics in Artificial Intelligence, Proceedings of the 9th European Conference*. *LNAI*, vol. 3229, Springer, Berlin, pp. 685–688.

Costantini S., Mostarda, L., Tocchio, A., Tsintza P. (2008). DALICA: agent-based ambient intelligence for cultural-heritage acenarios. *IEEE Intelligent Systems*, 23(2), pp. 34–41.

Damala, A., Cubaud, P., Bationo, A., Houlier, P., Marchal, I. (2008). Bridging the gap between the digital and the physical: design and evaluation of a mobile augmented reality guide for the museum visit. *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA '08)*, Athens, Greece, pp. 120–127.

Damiano, R., Galia, C., Lombardo, V. (2006). Virtual tours across different media in DramaTour project. *Workshop Intelligent Technologies for Cultural Heritage Exploitation at the 17th European Conference on Artificial Intelligence (ECAI 2006)*, Riva del Garda, pp. 21–25.

Ducatel K., Bogdanowicz M., Scapolo F., Leijten J., Burgelman J.-C. (2001). Scenarios for ambient intelligence in 2010. *Technical report, Information Society Technologies Programme of the European Union Commission (IST)*, Feb. 2001. http://www.cordis.lu/ist/.

Farella, E., Brunelli, D., Benini, L., Ricco, B., Bonfigli, M.E. (2005). Computing for interactive virtual heritage. *IEEE Multimedia*, 12(3), pp. 46–58.

Gentile A., Cossentino, M., Vitabile, S., Chella, A., Sorbello F. (2002). Intelligent agent mapping on a massively parallel MIMD computing platform. *Proceedings of 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2002)*, vol. II, CSREA Press, Las Vegas, pp. 850–855.

Giunchiglia F., Mylopoulos J., Perini A. (2002). The Tropos software development methodology: processes, models and diagrams. *AAMAS Conference 2002*.

Hodjat, B., Hodjat, S., Treadgold, N., Jonsson, I. (2006). CRUSE: a context reactive natural language mobile interface. *Proceedings of the 2nd Annual International Workshop on Wireless Internet (WICON '06)*, Boston, Massachusetts, 2–5 August 2006, ACM, New York, p. 20.

Huang H., Cerekovic, A., Pandzic, I. S., Nakano, Y., Nishida, T. (2009). Toward a multi-culture adaptive virtual tour guide agent with a modular approach. *AI & Society*, 24(3), pp. 225–235.

Ibanez, J., Aylett, R., Ruiz-Rodarte R. (2003). Storytelling in virtual environments form a virtual guide perspective. *Virtual Reality*, 7(1).

Jones, C.M., Lim, M.Y., Aylett, R. (2005). Empathic interaction with a virtual guide. *AISB Symposium on Empathic Interaction*, University of Hertfordshire, 12–15 April 2005.

Kling, R. 1991. Cooperation, coordination and control in computer-supported work. *Communications of the ACM*, 34(12), pp. 83–88.

Liarokapis F., Newman, R. M. (2007). Design experiences of multimodal mixed reality interfaces. *Proceedings of the 25th Annual ACM International Conference on Design of Communication*, El Paso, TX, USA, pp. 34–41.

Lind, J. (2001). Iterative software engineering for multiagent systems. *The MASSIVE Method. Lecture Notes in Computer Science*, vol. 1994, Springer, Berlin.

Lopez-Jaquero, V., Montero, F., Gonzalez, P. (2009). AB-HCI: an interface multi-agent system to support human-centred computing. *Software, IET*, 3(1), pp. 14–25.

McNeese, M.D.; Zaff, B.S.; Citera, M.; Brown, C.E.; Whitaker, R. (1995). AKADAM: eliciting user knowledge to support participatory ergonomics. *The International Journal of Industrial Ergonomics*, 15(5), pp. 345–363.

Nickerson, M. (2005). All the world is a museum: access to cultural heritage information anytime, anywhere. *Proceedings of International Cultural Heritage Informatics Meeting, ICHIM05*, Paris, September 2005, pp. 2–15.

Padgham, L., Winikoff, M. (2005). Prometheus: a practical agent-oriented methodology. *Agent-Oriented Methodologies*, edited by B. Henderson-Sellers and P. Giorgini, Idea Group.

Penserini, L., Liu, L., Mylopoulos, J., Panti, M., Spalazzi, L. (2003) Cooperation strategies for agent-based P2P systems. *WIAS: Web Intelligence and Agent Systems: An International Journal*, 1(1), pp. 3–21.

Pilato, G., Vitabile, S., Conti, V., Vassallo, G., Sorbello, F. (2004a). Web directories as a knowledge base to build a multi-agent system for information sharing. *Web Intelligence and Agent Systems, An International Journal*, 2(4), pp. 265–277.

Pilato, G., Vitabile, S., Conti, V., Vassallo, G., Sorbello, F. (2004b). A mobile agent based system for documents classification and retrieval. *Intelligenza Artificiale*, 1(3), pp. 34–40.

Rakotonirainy, A., Loke, S. W., Zaslavsky A. Multi-agent support for open mobile virtual communities. *Proceedings of the International Conference on Articial Intelligence (IC-AI 2000)* (Vol. I), Las Vegas, NV, USA, pp. 127–133, 2000.

Santoro, C., Paternò, F., Ricci G., Leporini B. (2007). A multimodal mobile museum guide for all. *Proceedings of the Mobile Interaction with the Real World (MIRW 2007) in Conjunction with the 5th Workshop on "HCI in Mobile Guides"*, Singapore, September 09.

Shah, N., Siddiqi, J. I. A., Akhgar, B. (2007). An intelligent agent based approach to MOSAICA's pedagogical framework. *Proceedings of the Ninth International Conference on Enterprise Information Systems (ICEIS)*, pp. 219–226.

Stock, O., Zancanaro, M., Busetta, P., Callaway, C., Krüger, A., Kruppa, M., Kuflik, V., Not, E., Rocchi, C. (2007). Adaptive, intelligent presentation of information for the museum visitor in PEACH. User Modeling and User-Adapted Interaction, 17(3), pp. 257–304.

Weiser M. (1991). The computer for the twenty-first century. *Scientific American*, pp. 94–10.

Chapter 12
Reinforced Operators in Fuzzy Clustering Systems

Andrei Doncescu, Sebastien Regis, and Nabil Kabbaj

Summary Knowledge-based systems need to deal with aggregation and fusion of data with uncertainty. To use many sources of information in numerical forms for the purpose of decision or conclusion, systems are supposed to have tools able to represent the knowledge in a mathematical form. One of the solutions is to use fuzzy logic operators.

12.1 Introduction

Modeling a nonlinear dynamic system can be achieved using differential equation or time series describing the behavior of the system based on the knowledge cause/effect. Although most identification methods assume that the input and output variables of the process are known, in reality it is often not clear which variables should be considered as input to the model. It seems obvious to introduce the knowledge about the system as background and to be able to deal with it in a fusion context. The most easy way is to express the knowledge in a linguistic form which mathematically expresses uncertainty. Several mathematical frameworks exist for modeling uncertainty:

1. probability theory,
2. belief theory,
3. fuzzy sets and possibility theory.

A. Doncescu (✉) · N. Kabbaj
LAAS-CNRS, University of Toulouse, 7, avenue du Colonel Roche, 31077 Toulouse, France
e-mail: adoncesc@laas.fr

N. Kabbaj
e-mail: nkabbaj@laas.fr

S. Regis
Grimaag-Guadeloupe, Campus de Fouillole, B.P. 592, 97157 Pointe Pitre Cedex, France
e-mail: sregis@univ-ag.fr

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,*
Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5_12, © Springer Science+Business Media, LLC 2010

The probability theory is the most commonly used. In this field, the a posteriori probabilities for an object to be a member of a class are computed according to Bayes decision rule. This probabilistic approach is criticizable for several reasons: initially the construction of a probability distribution requires much more information than an expert is able to provide. The choice of a parameterized family of distribution functions mainly results from a concern to simplify calculation. Consequently, the fit of the model to the expert opinion is debatable. An expert prefers to provide intervals rather than isolated values because his knowledge is of limited reliability and with accuracies. Belief theory handles inaccurate and uncertain information. The possibility theory and fuzzy sets are based on fuzzy logics. Fuzzy logics are characterized as "logics based on the real number." In these types of logic, one considers that the truth degrees are taken from the real line $\mathbb{R}$. Fuzzy modeling and control are typical examples of techniques that make use of human knowledge and deductive process basically using inference mechanisms. The advantage of fuzzy modeling is that the information can be either of numerical or of symbolic nature. Its representation as numerical degrees leads to a quantification of its characteristics (uncertain, imprecise, incomplete) which have to be taken into account in a fusion process. Therefore, the kernel of these mechanisms is the fusion operator defined as

$$F(x_1, x_2, x_3, \ldots, x_n), \tag{12.1}$$

where x_i denotes the representation of information issued from sensor i. The goal of the fusion operators (or aggregation) is to carry out the fusion of information resulting from various and varied sources, having the goal to take a better decision than from one source only, by reducing imprecision and uncertainty and increasing completeness. Therefore the fusion or aggregation is an reinforced information.

Information-fusion methods are executed in two ways:

1. combining each source in a parallel way: in this case, all data to be merged are available at the time of fusion, or
2. reviewing information: data are available at different moments, and the decision to be taken is updated as soon as new information comes in.

12.2 Fusion Operators

There is a great number of fusion operators. The choice and utilization of an aggregation operator depend on many parameters. This choice depends on the *fusion* itself. Before going further, it is important to give the definition of fusion suggested by Bloch and Hunter (2001):

> *The fusion consists to join together or aggregate the information coming from various sources, and to exploit this new resulting information in various applications like the decision, numerical estimation, etc.*

This definition points out two principal elements. First, the definition emphasizes the combination of the information. Then the accent is put on the fusion itself. Another

important aspect is to know what kind of data we seek to aggregate. There are two types of data: one is proposed by Bloch and Hunter (2001), and the other is presented by Dubois and Prade (2004). The latter data consist of the following:

– *The observations*. They describe the world from more or less particular point of view. We speak in general about numerical data provided by sensors.
– *The knowledge*. It describes how the world is *in general*. In this case we speak often about data from human observations rather than from sensors.
– *The preferences.* It is information which describes how we like the world to be. Of course, this information is coming from persons.
– *The regulations*. We speak about generic information presented on the form of laws.

According to the definition given in Dubois and Prade (1994) and Yager and Rybalov (1996), for any fusion operator F of two variables, one says that:

1. F is conjuctive if $F(x, y) \leq \min(x, y)$;
2. F is disjunctive if $F(x, y) \geq \max(x, y)$;
3. F behaves like a compromise if $x \leq F(x, y) \leq y$ if $x \leq y$ and $y \leq F(x, y) \leq x$ otherwise.

If all sources are reliable, it is possible to use a conjunctive fusion. But if some sources are reliable and some are not, or if their reliability is unknown, then it is better to use disjunctive fusion. A weighted logical combination can also be applied to merge data sources that have different degrees of reliability.

Bloch has been proposed the following classification to describe the operators in terms of their behavior (Bloch 1994):

1. Context-Independent Constant-Behavior (CICB) operators: they have the same behavior whatever the values of the information to combine.
2. Context-Independent Variable-Behavior (CIVB) operators: the behavior depends on the values of x and y.
3. Context-Dependent (CD) operators: they depend on a global knowledge or measure on the sources to be fusioned.

12.3 Fusion Operators in Fuzzy Sets and Possibility Theory

The operators used in these theories are CICB and are constituted by three families:

1. triangular norms T-norms $\rightarrow T$;
2. triangular conorms Γ-conorms $\rightarrow C$;
3. mean operators M.

A continuous T-norm is a continuous binary operation "$*$" on the real unit square which is:

1. commutative;
2. associative;

3. nondecreasing and having 1 for its unit element.

A continuity property is often added to these properties. T-norms generalize intersection to fuzzy sets. Examples of T-norms are:

1. minimum T-norm $x * y = \min(x, y)$, introduced by Dummett;
2. $x * y = \max(0, x + y - 1)$;
3. product T-norm $x * y = xy$.

It is easy to prove the following result: for any T-norm T, the following inequality holds:

$$\forall (x, y) \in [0, 1]^2, \quad T(x, y) \leq \min(x, y). \tag{12.2}$$

This result shows that the "min" is the greatest T-norm which has a conjunction behavior. This kind of operators is used when all the sources are reliable. We remark that in Probability and Dempster–Shafer, the operators are the product and the orthogonal sum, respectively, having a conjunction behavior and are CICB.

A T-conorm is defined as an operator $C : [0, 1]^2 \to [0, 1]$ such that C is commutative, associative, monotonic, and admitting 0 as unit element. It is easy to prove the following inequality for any T-conorm C:

$$\forall (x, y) \in [0, 1]^2, \quad C(x, y) \geq \max(x, y). \tag{12.3}$$

Disjunctive operators are used when at least one source is reliable. The other sources could be uncertain. The most known is the maximum.

A particular class is the uninorms proposed by Yager and Rybalov (1996). They are commutative, associative, and having a neutral element $e \in [0, 1]$, which the user may fix. In practice, an uninorm is often defined by a T-norm on the interval $[0, e]$ and by a T-conorm on the interval $[e, 1]$. We point out that some uninorms are symmetrical sums (Dubois and Prade 2004). It is pointed out that the symmetrical sums, introduced by Silvert (1979), are operators whose characteristic is to be symmetrical concerning a subset and its complement. Similar to the uninorms, the hybrid operators are the combinations of T-norms and T-conorms named mixed connectives. The goal of these operators is to get the advantages of T-norms and T-conorms by the variation of a parameter. These operators have been studied by Zimmerman and Zynso (1980) and later by Piera-Carreté et al. (1988). The most known are:

– the linear connective:

$$\alpha T(x_1, \ldots, x_n) + (1 - \alpha) C(x_1, \ldots, x_n); \tag{12.4}$$

– the geometrical connective:

$$T(x_1, \ldots, x_n)^{\alpha} + C(x_1, \ldots, x_n)^{(1-\alpha)}, \tag{12.5}$$

where $0 \leq \alpha \leq 1$.

The zero norms (*nullnorms*) defined by Calvo et al. (2001) are commutative, associative operators, having an absorbent element $a \in [0, 1]$, which the user can fix a priori.

The means operators provide a value between the minimum and the possible maximum. A mean operator M is defined as a function such that:

1. $\min(x, y) \leq M(x, y) \leq \max(x, y)$;
2. $M(x, y) = M(y, x)$;
3. M is increasing w.r.t. both arguments.

We notice that the aggregation operators need to satisfy the monotonic condition. We could interpret this condition by the fact that if the marginal information increases, the numerical value also increases or, in a less strong condition, does not decrease.

Let us quote, for example, the arithmetic and geometrical means or the median. Ordered weighted averages suggested by Yager (1988) (*Ordered Weighted Averaging: OWA*) are also mean operators into which it is possible to introduce a weighting depending on the importance and on the reliability of the sources. The class of operators OWA has the advantage of being stable reporting to positive linear transformation. The Min and Max are particular cases of OWA operators, but the most important property of these operators is the ability to represent optimistic or pessimistic attitude.

12.4 The Triple Π Operator

Suppose that for a given class, a vector form has membership degrees important for all features considered. In the human reasoning, an aggregation of all this marginal information will be higher than each degree taken separately (Yager and Rybalov 1998). In this type of reasoning, the membership degrees which are strong will be reinforced mutually. This behavior is called *positive reinforcement*. Of a similar reasoning, if for a given class, an object has small membership degree, the aggregation will be weaker than weakest of the membership degree values. We speak in this case about *negative reinforcement*.

The total reinforcement is a property which translates certain aspects of the human reasoning. Using the operator having this property can thus be interesting in measurement where we seek a system close to this type of reasoning. The completely reinforced operators are a particular class of operators having the characteristic to be both positively and negatively reinforced. The concept of reinforcement was presented by Yager and Rybalov (1998). The only completely reinforced operator that we know is the triple Π developed by these two authors (Yager and Rybalov 1998). We figure out that this triple Π is also a symmetrical sum.

Definition 12.1 An aggregation operator L whose arguments are within the interval $[0, 1]$ has the property of positive reinforcement if when all its attributes are affirmative (i.e., greater than or equal to 0.5), it satisfies the conditions

$$L(x_1, \ldots, x_n) \geq \max_i \big[L(x_i)\big]. \tag{12.6}$$

Similarly, an aggregation operator L whose arguments are within the interval [0, 1] has the property of negative reinforcement if when all its attributes are nonaffirmative (i.e. lower than or equal to 0.5), it satisfies

$$L(x_1, \ldots, x_n) \leq \min_i \big[L(x_i)\big]. \tag{12.7}$$

An operator having the above two properties is defined as being *totally reinforced (fully reinforced)*.

The T-norms are negative reinforcement operators ($T(x_1, \ldots, x_n) \leq \min_i[T(x_i)]$), but they are not positive reinforced. In addition, the T-conorms are positive reinforced ($C(x_1, \ldots, x_n) \geq \max_i[C(x_i)]$), but they are not negative reinforced. We could hope that combinations of T-norms and T-conorms (as connective mixed ones) are completely reinforced, but Yager and Rybalov found counterexamples proving that this is not true (Yager and Rybalov 1998).

The mean operators are not positively reinforced or reinforced negatively by definition. Indeed, for an average, $\min_i(x_i) \leq M(x_1, \ldots, x_n) \leq \max_i(x_i)$.

The only operator which is (to our knowledge) completely reinforced is the triple Π defined by Yager and Rybalov (1998):

$$PI(x_1, \ldots, x_n) = \frac{\Pi_{j=1}^{n} x_j}{\Pi_{j=1}^{n} x_j + \Pi_{j=1}^{n}(1 - x_j)}. \tag{12.8}$$

Recall that this operator is also a symmetrical sum (Silvert 1979). It is also to be noted that the symmetric sum of two fuzzy sets has the property that the sum of complements is the complement of the sum. We must also note that there are several studies and works on the general properties of symmetrical sums (Silvert 1979; Dubois et al. 1993), but there are few works done on the differences between symmetrical sums.

The property of the total reinforcement is thus particularly interesting because it makes it possible to obtain a good modeling of the human behavior, which is often the goal of many knowledge-based systems. It should be noted that the triple Π incorporates information of the type of the *observations in order to refine the information related to the real world* (Bloch and Hunter 2001) and can be used in this type of information fusion.

12.5 The *Mean Triple Π*

12.5.1 The Mean Triple Π

Although the triple Π is an interesting operator because of the fact that it is completely reinforced, it is sometimes more judicious to use operators of the type *mean*. As first underlined by Yager (1996) and then by Bloch and Hunter (2001) and

Dubois and Prade (2004), when the signals are used to represent the same phenomenon (these signals may be independent of others or not), it is more relevant, from a conceptual point of view, to use a mean operator in order to synthesize the information.

The basic idea which leads to the definition of this new mean operator is to seek a mean operator which has properties close to those of the triple Π. This means that

$$PI(x_1, \ldots, x_n) = \frac{\Pi_{j=1}^{n} G(x_j)}{\Pi_{j=1}^{n} G(x_j) + \Pi_{j=1}^{n} G(1 - x_j))}, \tag{12.9}$$

where $G(x)$ is a function named *generatrix function* which is positive and increasing (Waissman-Vilanova 2000; Silvert 1979). To obtain the idempotent property, we have considered the function $G(x) = x^{1/n}$, where n is the dimension of the vector x. We can define a new aggregation operator,

$$MPI(x_1, \ldots, x_n) = \frac{\Pi_{j=1}^{n} (x_j)^{(1/n)}}{\Pi_{j=1}^{n} (x_j)^{(1/n)} + \Pi_{j=1}^{n} (1 - x_j)^{(1/n)}} \tag{12.10}$$

$$= \frac{1}{1 + \Pi_{j=1}^{n} [\frac{1-x_j}{x_j}]^{1/n}}. \tag{12.11}$$

We call this new operator *mean triple* Π, by reference to the triple Π from which it is obtained.

Proposition 12.1 *The mean triple* Π *defined above is a mean operator.*

Proof We show that this operator is a mean operator by checking the properties of the mean operators (see Yager 1996):

1. the commutativity: $MPI(x, y) = MPI(y, x)$;
2. the monotonicity: $MPI(x, y) \geq MPI(z, t)$ if $x \geq z$ and $y \geq t$;
3. the idempotency: $MPI(x, \ldots, x) = x$;
4. the self-identity : $MPI[B, \langle MPI(B)\rangle] = MPI(B)$.

The first three conditions can be deduced easily from the properties of the product function and n-square function. The most difficult property is the self-identity shown below. We want to demonstrate that:

$$MPI\big(x_1, \ldots, x_n, MPI(x_1, \ldots, x_n)\big) = MPI(x_1, \ldots, x_n, MPI) = MPI(x_1, \ldots, x_n).$$

Therefore,

$$MPI(x_1, \ldots, x_n, MPI) = \frac{\Pi_{j=1}^{n} (x_j)^{1/(n+1)} \times (MPI)^{1/(n+1)}}{D}$$

with

$$D = \left[\Pi_{j=1}^{n} (x_j)^{1/(n+1)} \times (MPI)^{1/(n+1)} + \Pi_{j=1}^{n} (1 - x_j)^{1/(n+1)} \times (1 - MPI)^{1/(n+1)}\right]$$

and

$$(MPI)^{1/(n+1)} = \left(\frac{\Pi_{j=1}^{n}(x_j)^{(1/n)}}{\Pi_{j=1}^{n}(x_j)^{(1/n)} + \Pi_{j=1}^{n}(1-x_j)^{(1/n)}} \right)^{1/(n+1)}$$
$$= \frac{\Pi_{j=1}^{n}(x_j)^{1/n(n+1)}}{(\Pi_{j=1}^{n}(x_j)^{(1/n)} + \Pi_{j=1}^{n}(1-x_j)^{(1/n)})^{1/(n+1)}}.$$

By simplification with

$$\frac{1}{(\Pi_{j=1}^{n}(x_j)^{(1/n)} + \Pi_{j=1}^{n}(1-x_j)^{(1/n)})^{1/(n+1)}}$$

we have

$$MPI(x_1, \ldots, x_n, MPI)$$
$$= \frac{\Pi_{j=1}^{n}(x_j)^{[1/(n+1)+1/n(n+1)}]}{\Pi_{j=1}^{n}(x_j)^{[1/(n+1)+1/n(n+1)]} + \Pi_{j=1}^{n}(1-x_j)^{[1/(n+1)+1/n(n+1)]}}$$
$$= \frac{\Pi_{j=1}^{n}(x_j)^{[(n+1)/n(n+1)]}}{\Pi_{j=1}^{n}(x_j)^{[(n+1)/n(n+1)]} + \Pi_{j=1}^{n}(1-x_j)^{[(n+1)/n(n+1)]}}$$
$$= \frac{\Pi_{j=1}^{n}(x_j)^{(1/n)}}{\Pi_{j=1}^{n}(x_j)^{(1/n)} + \Pi_{j=1}^{n}(1-x_j)^{(1/n)}}$$
$$= MPI = MPI(x_1, \ldots, x_n). \qquad \square$$

The mean triple Π is a new mean operator of aggregation obtained from the triple Π. It is obvious that the mean triple Π cannot be completely reinforced since by definition it is a mean: the numerical value is between the maximum and minimum. However, it has a property similar to the total reinforcement of triple Π, which is based on a comparison with the classic arithmetic mean. The definition of the property is given below.

12.5.2 The Mean Reinforcement

Property 12.1 *Let MPI be the mean triple Π. We consider the classic arithmetic mean $\frac{1}{n}\sum_{j=1}^{n} x_j$. Then: If $x_j \geq 0.5$, $j \in 1, \ldots, n$, then we have*

$$MPI(x_1, \ldots, x_n) \geq \frac{1}{n}\sum_{j=1}^{n} x_j; \tag{12.12}$$

if $x_j \leq 0.5$, $j \in 1, \ldots, n$, *then we have*

$$MPI(x_1, \ldots, x_n) \leq \frac{1}{n} \sum_{j=1}^{n} x_j. \tag{12.13}$$

This property is called mean reinforcement by reference to the total reinforcement of the triple Π.

Proof First, let $x_j \geq 0.5$ for all $j \in 1, \ldots, n$. We have to show that

$$\frac{\Pi_{j=1}^{n}(x_j)^{(1/n)}}{\Pi_{j=1}^{n}(x_j)^{(1/n)} + \Pi_{j=1}^{n}(1-x_j)^{(1/n)}} \geq \frac{1}{n} \sum_{j=1}^{n} x_j.$$

This is equivalent to

$$\begin{aligned}
\frac{\Pi_{j=1}^{n}(x_j)^{(1/n)} + \Pi_{j=1}^{n}(1-x_j)^{(1/n)}}{\Pi_{j=1}^{n}(x_j)^{(1/n)}} &\leq \frac{n}{\sum_{j=1}^{n} x_j}, \\
1 + \frac{\Pi_{j=1}^{n}(1-x_j)^{(1/n)}}{\Pi_{j=1}^{n}(x_j)^{(1/n)}} &\leq \frac{n}{\sum_{j=1}^{n} x_j}, \\
\frac{\Pi_{j=1}^{n}(1-x_j)^{(1/n)}}{\Pi_{j=1}^{n}(x_j)^{(1/n)}} &\leq \frac{n}{\sum_{j=1}^{n} x_j} - 1, \\
\left(\prod_{i=1}^{n}\left(\frac{1}{x_i} - 1\right)\right)^{\frac{1}{n}} &\leq \frac{1}{\frac{\sum_{i=1}^{n} x_i}{n}} - 1,
\end{aligned}$$

and, finally, by taking the logarithms, to

$$\frac{1}{n} \sum_{i=1}^{n} \ln\left(\frac{1}{x_i} - 1\right) \leq \ln\left(\frac{1}{\frac{\sum_{i=1}^{n} x_i}{n}} - 1\right).$$

The second derivative of the function $f(x) = \ln(\frac{1}{x} - 1)$ is

$$\frac{d^2 f(x)}{dx^2} = \frac{1 - 2x}{(x^2 - x)^2}.$$

The inequality above follows, by the Jensen inequality on the interval [0.5, 1], from the concavity of f within interval [0.5, 1], where the second derivative of f is negative: $\frac{d^2 f(x)}{dx^2} \leq 0$.

The second inequality follows similarly from the fact that f is convex on $[0, 0.5]$. □

This property translates the fact that when the signals are in accord, the average triple Π discriminates the classes better that the arithmetic mean. This property

can be interesting when we want to evaluate the correlations between different the sources.

12.6 LAMDA Clustering System

The clustering methods deal with unsupervised classification. The term cluster analysis was coined by Tryon in 1939 that encompasses a number of methods and algorithms for categorizing objects of similar kinds. The main objective of clustering is organize a collection of data items into some meaningful clusters, so that items within a cluster are more similar to each other than they are to items in the other clusters. A class or cluster is a set of similar objects (having similar characteristics). Clustering (or segmentation) of objects starts with an arbitrary choice of a similarity measure that describes proximity between different objects. The choice of the similarity (or dissimilarity/distance) measure ultimately defines the outcome of the clustering and is far more important than the choice of the actual clustering algorithm. In general, any similarity measure can be converted into a dissimilarity measure by applying a suitable decreasing function. Some clustering algorithms assume that the dissimilarity or distance measure is a metric.

LAMDA is a fuzzy methodology of conceptual clustering and classification based on the concept of adequacy to each class that replaces the usual "distance to a center" approach. Moreover, the class adequacy concept is expressed as the "fuzzy" truth value of a compound sentence using logical connectives between elementary assertions. We would like to point out that LAMDA method treats objects in a sequential manner. By its value, each descriptor contributes to the global adequacy of one object to one class through marginal adequacy degree (MAD). We use a fuzzy logic operator which interpolates between union and intersection with an adjustable parameter called "exigency." To make a direct confrontation between classes and objects possible, it is necessary that the concept be described with the same descriptor as the one used for observations. Each object is described by a set of attributes or descriptors and represented by vectors of n components. In LAMDA method, descriptors can be considered qualitative or quantitative.

The main properties of LAMDA are the following:

1. both supervised and unsupervised learning may be carried out,
2. simultaneous processing of numerical and qualitative information,
3. learning is performed in a sequential and incremental manner,
4. classification algorithms are based on linear compensated hybrid connectives which aggregate the marginal adequacy degrees (*MAD*) to obtain the global adequacy degree (*GAD*) of an object to a class,
5. total indistinguishability (chaotic homogeneity) in the description space is modeled by means of a special class called the Non-Informative Class (*NIC*); this class accepts any objects with the same adequacy degree and thus naturally introduces a classification threshold,
6. possibility to obtain different classifications from the same group of objects by means of the "exigency" concept.

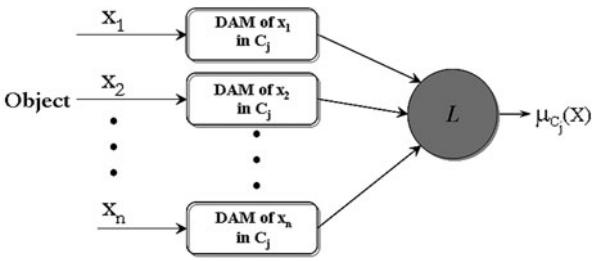


Fig. 12.1 The LAMDA structure

The input data for this algorithm is a collection of objects (individuals or observations), described by a set of n qualitative or quantitative descriptors (attributes), represented as vectors $\mathbf{x}$, where the jth component is the value taken by the jth descriptor if it is numeric or symbolic. In the case of a qualitative descriptor, this value is called modality. The information carried by each descriptor contributes to the membership of the element to the class by means of the Marginal Adequacy Degree (*MAD*) (see Fig. 12.1). LAMDA methodology can be split in 3 stages:

1. Compute the membership function of each source named Marginal Adequacy Degree *MAD*. This membership function is similar to an a posteriori Bayesian probability.
2. The assignment of any object to a class is computed by fusion of all marginal information available *MAD* using a fuzzy fusion operator, the result being named *GAD* (Global Adequacy Degree). We remark that $MAD = 1$ represents the total adequacy of the given attribute to the class and $MAD = 0$ represents its total inadequacy.
3. Whenever the *NIC* class corresponds to the maximum *MAD*, the object is considered as unrecognized, and no known concept is related to it. Two alternatives exist:
   a. the object is merely ignored, or unclassified;
   b. the object is considered to belong to an unknown concept, and it will be taken as the first element of a new class. This case is called self-learning. The new concept (class) will be initialized by this object and by the parameters of the *NIC*.

The marginal adequacy degree function, chosen in this application, is a fuzzy interpolation of the binomial probability:

$$MAD_{j,k} = \rho_{j,k}^{\delta(x_j, c_{j,k})} (1 - \rho_{j,k})^{1-\delta(x_j, c_{j,k})}, \tag{12.14}$$

where $\delta(x_j, c_{j,k})$ is a distance toward a central parameter of class C_k. To estimate the parameters $\rho_{j,k}$ and $c_{j,k}$ for a given learning data set, the minimization of a likelihood criterion is used:

$$J(\rho, c) = \max_{\rho,c} \left| MAD(x_i)_{i=1}^{n} \right|. \tag{12.15}$$

This is obtained for each descriptor (the index of descriptor is omitted) by

$$\rho_k = \frac{1}{n} \sum_{i=1}^{n} \sigma(x_i, c_{i,k}) \tag{12.16}$$

and resolving

$$\sum_{i=1}^{n} \frac{\partial}{\partial c_{i,k}} \delta(x_i, c_{i,k}) = 0. \tag{12.17}$$

The distance used here is the scalar Euclidean $\delta(x_i, c_{i,k}) = |x_i, c_{i,k}|$.

It must be noted that in order to calculate the adequacy of an element to a class, both must have the same descriptor set. Then, all the *MAD* are aggregated in order to obtain a Global Adequacy Degree (*GAD*) of the object to the class. This is made by a convex interpolation of fuzzy logic connectives L_α the Mixed Connective of linear compensation, presented before, which in the new notation become

$$\begin{aligned} &GAD(MAD_1, MAD_n) \\ &\quad = \alpha * T(MAD_1, MAD_n) + (1 - \alpha) * C(MAD_1, MAD_n), \end{aligned} \tag{12.18}$$

where MAD_i is the marginal adequacy of the object and $\alpha \in [0, 1]$, to be coherent with Fuzzy Logic aspects that include compatibility with Boolean Logic; T is an iterated T-norm, and C its dual T-conorm with respect to the negation (complement to 1). The parameter α is called the *Exigency Index*, and it is possible to associate different classifications to the same data set, depending on the value chosen for α. As shown in Piera-Carreté et al. (1988), recognition is more exigent as α increases; therefore, there will be more nonrecognized objects. Similarly, if α increases, learning becomes more selective (or exigent) as the number of objects assigned to the *NIC* increases, and so does the number of created classes. Thus, by changing the value of α, different partitions from the same data set, based on the same logical criterion, can be obtained.

The clustering algorithm LAMDA is presented below:

1. Get the extremal values $x_{i,\min}$ and $x_{i,\max}$ and those of the quantitative components. Replace the value x_i by its normalized value for each descriptor i:

$$x_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}. \tag{12.19}$$

2. Compute the marginal adequacy degrees for each descriptor which are respectively MAD_i, $i = 0, \ldots, d$.
3. Compute the global adequacy. Search for the maximum *GAD* degree to assign the object to a class.
4. Two cases may occur:
   a. Recognition mode. Object x is placed in C_i. If C_i is *NIC*, then object x is said to be "unrecognized."

b. Self-learning, or Concept Formation mode. There are two possibilities:
   i. GAD_i does not correspond to *NIC*; in this case, x is placed in C_i, and the parameters of C_i will be modified to include x.
   ii. GAD_i corresponds to *NIC*. This means that x is the first element of a new class C_{k+1}, and the representation of this new class will depend on x.

For the modification of class parameters, we use the following algorithm:

$$c_{i,k} = c_{i,k} + \frac{x_i - c_{i,k}}{N_{NIC} + 1} \tag{12.20}$$

and

$$\rho_{i,k} = \rho_{i,k} + \frac{\delta(x_i, c_{i,k}) - \rho_{i,k}}{N_{NIC} + N_k + 1}, \tag{12.21}$$

where N_k is the number of elements assigned to class C_k, and N_{NIC} is a virtual number of elements of the *NIC* class; it is a parameter introduced in order to initialize the new classes whenever self-learning is applied. It can be noticed that $\rho_{i,NIC} = \frac{1}{2}$, so that the initial parameters of the new class are

$$\rho_{i,k} = \frac{1}{2}\big(N_{NIC} + \delta(x_i, c_{i,k})\big)\frac{1}{N_{NIC} + 1} \tag{12.22}$$

and $c_{i,k} = x_i$.

12.7 Experimental Results

Biological knowledge is inherently incomplete, owing to the complexity of living systems and the limitation of scientific methods available for the study of those systems. The incompleteness of knowledge constantly manifests itself unexplained observations. To account for these novel observations, biologists need to revise or extent the existing knowledge. The application that we treat relates to fusion of information during a processes of classification in a biotechnology. We need to fusion information resulting from various measured biochemical parameters (pH, dioxygene, carbon dioxide, etc.) allowing one to carry out a nonsupervised classification. This classification is based on the hypothesis that the measurements expressed the same biological phenomena. Therefore, the obtained classes must correspond to the physiological states of these microorganisms (note that we have *information from real world*). The physiological state is the biological reaction inside the microorganisms which has, for consequence, the production of a specific metabolite or the reproduction of the cells. In this application the physiological states are known based on the analysis of respiratory quotient. The goal of the nonsupervised classification is to determine the class corresponding to the physiological states without any knowledge. We have noticed that the Mixed Connective leads to dissatisfaction in the classification processes. Therefore, we have replaced the Mixed Connective by triple Π and the Mean triple Π.

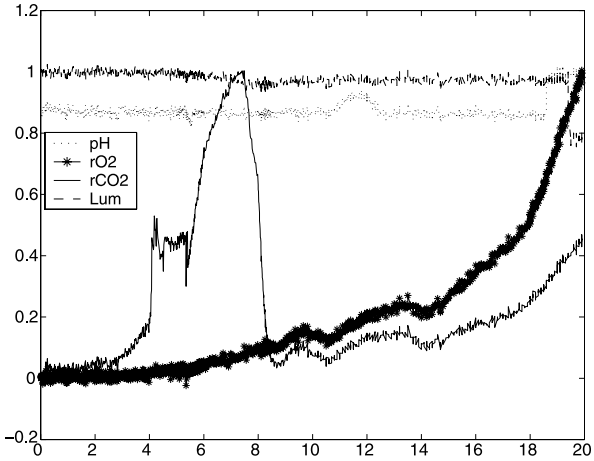


Fig. 12.2 The four biochemical parameters with noise (SNR = 40 dB)

Table 12.1 Comparison of the classification using the mean triple Π (*MPI*) and the triple Π (*PI*). CTN signifies that the Classification is Too Noisy to obtain a significant result

	% classif. M3PI	% classif. 3PI
SNR = 46.02 dB	13.47%	CTN
SNR = 40 dB	11.56%	CTN
SNR = 30.40 dB	CTN	CTN

In comparison to triple Π, the Mean triple Π provides a ranging value between the minimum and the maximum of the marginal membership degrees, making the synthesis between the various manifestations of the same event. In practice, we noted that the results of classification of the two operators were generally similar.

A notable exception between triple Π and the Mean triple Π is the classification of the noisy signals. The mean triple Π due to its property of smoothing is more robust with the noisy signal than triple Π. On Fig. 12.2, four parameters (pH, rO_2, rO_2, and Luminance) have been used. The two operators were tested on these noisy signals, and two classifications are disturbed by the presence of noise. Nevertheless, even in the presence of noise, the classification using the average triple Π provides at least a class which characterizes the fermentation state (state 1); see Table 12.1.

12.8 Uncertainties and Maximum of Modulus of Wavelet Transform

By uncertainties we mean that a transition between two classes is not well defined (a class overlaps the other) or that an isolated point belonging to a class is located among a set of other points belonging to another class. These uncertainties come from the perturbations due to measurement noise or from the analyzed image itself.

The classification is perturbed, and the borders between classes are not always well defined. Moreover, the lack of meaningful data reinforces such uncertainties in classification. To deal with the noise problem, one can use filters, but the result is linked to the nature of the filter, and we do not really know if the filtering causes the loss of meaningful and pertinent information. That is why we propose to use the maximum of modulus of wavelet transform.

12.9 Classification and Maximum of Modulus of Wavelet Transform

12.9.1 The Continuous Wavelet Transform

Wavelet theory is experiencing an increasing success, and wavelets are now being used in many fields. The wavelets have time-scale properties that are very interesting for the analysis of nonstationary signals. As we said above, we suppose that the singular points or the inflexion points may correspond to the transition between two classes and, consequently, between two different regions. To detect those points, we use the Maximum of Modulus of Wavelet Transform (Mallat 1991; Mallat and Zhong 1992). The main particularity of this Maximum of Wavelet Transform is using wavelet which is the first or second derivative of smoothing function (Gaussian for example):

$$\psi(t) = \frac{d\theta(t)}{dt}. \tag{12.23}$$

The wavelets are a powerful mathematical tool of nonstationary signal analysis (of signals whose frequencies change with time). Unlike the Fourier Transform, Wavelet Transform can provide the time-scale localization. The performance of the Wavelet Transform is better than that of the windowed Fourier Transform. Because of these characteristics, Wavelet Transform can be used for analyzing nonstationary signals such as transient signals. Wavelets Transformation (WT) is a rather simple mechanism used to decompose a function into a set of coefficients depending on scale and location. The definition of the Wavelets Transform is

$$W_{s,u} f(x) = (f \star \psi_{s,u})(x) = \int f(x)\psi\left(\frac{x-u}{s}\right)dx, \tag{12.24}$$

where ψ is the wavelet, f is the signal, $s \in R^{+*}$ is the scale (or resolution) parameter, and $u \in R$ is the translation parameter. The scale plays the role of frequency. The choice of the wavelet ψ is often a complicated task. We assume that we are working with an admissible real-valued wavelet ψ with r vanishing moments ($r \in N^*$).

The wavelet is translated and dilated as in the following relation:

$$\psi_{u,s} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right). \tag{12.25}$$

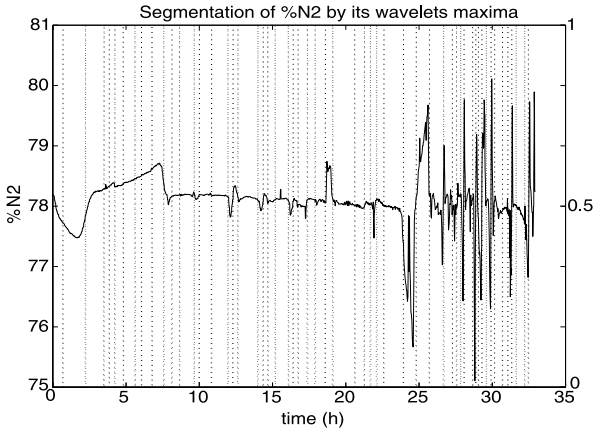


Fig. 12.3 Segmentation of N2 (nitrogen). Each vertical dotted line corresponds to a singularity of the signal detected by wavelets. The wavelet is a DOG (the first derivative of Gaussian), and the scales go from 2^0 to 2^3

The dilation allows the convolution of the analyzed signal with different sizes of "window" wavelet function. For the detection of the singularities and of the inflexion points of the biochemical signal, we use the Maxima of Modulus of Wavelets Transform (Mallat and Hwang 1992). The idea is to follow the local maxima at different scales and to propagate from low to high frequencies. These maxima correspond to singularities, particularly when the wavelet is the derivative of a smooth function,

$$\psi(x) = \frac{d\theta(x)}{dx},$$

$$W_{s,u} f(x) = f * \psi_{s,u} = f(x) * \frac{d\theta(x/s)}{dx}.$$

Yuille and Poggio (1986) have shown that if the wavelet is the derivative of the Gaussian, then the maxima belong to connected curves that are continuous from one scale to another. The detection of the singularities of the signal is thus possible by using the wavelets (see, for example, Fig. 12.3).

The discretization form of Continuous Wavelet Transform is based on the following form of the Mother Wavelet:

$$\psi^{m,n}(t) = a_0^{-m/2} \psi\left(\frac{t - nb_0 a_0^m}{a_0^m}\right). \tag{12.26}$$

By selecting a_0 and b_0 properly, the dilated mother wavelet constitutes an orthonormal basis of $L^2(R)$. For example, the selection of $a_0 = 2$ and $b_0 = 1$ provides a dyadic-orthonormal Wavelet Transform (DWT). The decomposed signals by *DWT* will have no redundant information thanks to the orthonormal basis.

Jiang et al. (2003) have proposed to select the maxima by using thresholding. Besides, all the singularities are not relevant; only some of them are meaningful.

However, as stated above, the thresholds proposed by Jiang et al. are chosen empirically. To select the meaningful singularities, we propose using the Hölder exponent. The Hölder exponent is a mathematical value allowing characterization singularities. The fractal dimension could also be used, but only the Hölder exponent can characterize locally each singularity. A singularity at a point x_0 is characterized by the Hölder exponent (also called the Hölder coefficient or Lipschitz exponent). This exponent is defined as the most important exponent α allowing us to verify the inequality

$$\left|f(x) - P_n(x - x_0)\right| \leq C|x - x_0|^{\alpha(x_0)}. \tag{12.27}$$

We must remark that $P_n(x - x_0)$ is the Taylor Development and basically $n \leq \alpha(x_0) < n + 1$. The Hölder exponent measures the remainder of a Taylor expansion and, moreover, measures the local differentiability:

1. $\alpha \geq 1$: $f(t)$ is continuous and differentiable.
2. $0 < \alpha < 1$: $f(t)$ is continuous but nondifferentiable.
3. $-1 < \alpha \leq 0$: $f(t)$ is discontinuous and nondifferentiable.
4. $\alpha \leq -1$: $f(t)$ is no longer locally integrable.

Therefore, the Hölder exponent can be extended to a distribution. For example, the Hölder exponent of the Dirac function is equal to -1. A simple computation leads to a very interesting result of the Wavelets Transform (Jaffard 1997):

$$\left|W_{s,u} f(x)\right| \simeq s^{\alpha(x_0)}. \tag{12.28}$$

This relation is remarkable because it allows one to measure the Hölder exponent using the behavior of the Wavelets Transform. Therefore, at a given scale $a = 2^N$, the $W_{a,b} f(x)$ will be the maximum in a neighborhood of the signal singularities. The detection of the Hölder coefficient is linked to the vanishing moment of the wavelet: if n is the vanishing moment of the wavelet, then it can detect the Hölder coefficients less than n (Mallat and Hwang 1992). We use a *DOG* wavelet (DOG: the first derivative of Gaussian) with a vanishing moment equal to 1; consequently, we can only detect the Hölder coefficients smaller than 1. This is not a real problem because we are interested (*in this application*[1]) by the singularities such as step or Dirac, and the Hölder coefficients of these singularities are smaller than 1. The values of the used integers are not meaningful: they are consecutive, and the only obligation is that an integer corresponds to one and only one class. The functions by stage are introduced in the classification as a new descriptor of the object. Besides, the maximum of modulus wavelet transform and, obviously, the functions by stage represent the borders of the different regions in the images. The functions by stage make it possible to have more precise borders in the LAMDA classification, as we will see in the next section.

[1]However, for others applications in bioprocesses, it is always possible to use other wavelets with greater vanishing moments.

12.9.2 Maximum into the Classification

The function by stage enables us to influence the classification of LAMDA. Because of the fact that the *GAD* triple Π is a total reinforced operator (Yager and Rybalov 1998) (unlikely the T-norm which is only negatively reinforced), LAMDA tends to facilitate the functions by intervals in comparison with the other data (here, the biochemical parameters) providing that these functions by stages are meaningful. Let us demonstrate this assertion.

Proof The triple Π is a reinforced operator, that is, it is positively reinforced and negatively reinforced. Particularly, the triple Π is positively reinforced, i.e., if all the descriptors are affirmative (that is, all the *MAD* are higher than 0.5), then we have

$$GAD_{j,i}(x_1, \ldots, x_n) \geq \max_{i=1,\ldots,n} \left[MAD_{j,i}(x_i)\right] \tag{12.29}$$

□

Let us take one meaningful stage of the function by stage, this function by stage being the lth descriptor of the objects to classify, and the integer distinguishing this stage being noted A. The class featuring by this particular stage is called C_s. This stage is meaningful, and it agrees with the *MAD* of the line of level gray in the temporal interval where the stage is defined. Let us suppose that all those *MAD* are *affirmative*. For all the objects that are located in the temporal interval where the stage is defined, the *MAD* for this stage is maximum. Besides, for an object located in this interval, the distance between the lth descriptor of the object which corresponds to this function by stages and the lth component of the center (called $c_{s,l}$) is equal to zero as we have

$$\alpha(x_l, c_{s,l}) = \alpha(A, A) = 0. \tag{12.30}$$

By consequence the *MAD* for the descriptor l is maximum:

$$MAD_s(x_l) = MAD_{s,l} = \rho_{sl}^{1-\alpha(x_l, c_{s,l})}(1 - \rho_{sl})^{\alpha(x_l, c_{s,l})} = \rho_{sl}. \tag{12.31}$$

Thus, according to (12.29), the *GAD* will tend to be very important for this class. In fact, the *GAD* will be equal to one because of the value of ρ_{sl} (one can easily show that here $\rho_{sl} = 1$ according to (12.16)) as we can see in the following equation:

$$MAD_{s,l} = \rho_{sl} = 1;$$

then

$$GAD_s(x_1, \ldots, x_l, \ldots, x_n) \geq MAD_{s,l} = 1;$$

and then

$$GAD_s(x_1, \ldots, x_l, \ldots, x_n) = 1.$$

If we compute the *MAD* for the other stages (more precisely, for the classes featuring by those classes $C_{r,r\neq s}$), they will not always be as optimal as the *MAD* of the class C_s since the values characterizing those stages are all different of A, and thus the distance between the lth component of the object and the lth component of the center of each class will always be greater than zero. For those classes, the *MAD* will not be as important as for the class C_s, i.e.,

$$GAD_{r,r\neq s}(x_1,\ldots,x_l,\ldots,x_n) \leq 1 = GAD_s(x_1,\ldots,x_l,\ldots,x_n).$$

So the maximum provides implicitly the borders of the classes and thus helps the region detection.

12.10 Conclusion

According to the analysis made up in the first section of the chapter, for the sources of information describing the same phenomena, it is better to use a mean operator. The results of nonsupervised time series classification obtained using 3Π and mean triple Π have shown that there are no revelent differences between these operators. The explication is based on the fact that the sources are in perfect concordance; therefore, the reinforced operator and the mean operators give very close resulting scores. Of course, the new operator of aggregation mean triples Π seems well adapted to analyze the time series describing the strong nonlinearity of biological systems. This operator combines the properties of the completely reinforced operators and the mean operators. The results obtained on the time series show that this operator is less sensitive to the noise than the operator of Yager and Rybalov (1998). Therefore, in the case of biological signals which have an important response time and are noisy, we can use it as an aggregation operator. In the future, we want to compare the classification obtained by triple Π and mean triple Π in the goal of introducing the notion of pertinence of data in biological system analysis.

References

I. Bloch, Information combination operators for data fusion: a comparative fusion with classification. Technical report, ENST, Paris, 1994. Rapport Technique.

I. Bloch and A. Hunter, Fusion: general concepts and characteristics. *International Journal of Intelligent Systems*, 16:1107–1134, 2001.

T. Calvo, B. De Baets, and J. Fodor, The functional equations of Frank and Alsina for uninorms and nullnorms. *Fuzzy Sets and Systems*, 120:385–394, 2001.

D. Dubois and H. Prade, La fusion d'informations imprécises. *Traitement du Signal*, 11(6):447–458, 1994.

D. Dubois and H. Prade, On the use of aggregation operations in information fusion process. *Fuzzy Sets and Systems*, 142:143–161, 2004.

D. Dubois, H. Prade, and R. R. Yager, editors, *Readings in Fuzzy Sets for Intelligent Systems*. Morgan Kaufman, San Mateo, 1993.

S. Jaffard (1997), Multifractal formalism for functions part 1 and 2. *SIAM Journal of Mathematical Analysis*, 28(4):944–998.

T. Jiang, B. Chen, X. He, and P. Stuart (2003), Application of steady-state detection method based on wavelet transform. *Computer and Chemical Engineering*, 27(4):569–578.

S. Mallat, Zero crossing of a wavelet transform. *IEEE Transactions on Information Theory*, 37:1019–1033, 1991.

S. Mallat and W.-L. Hwang (1992), Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643.

S. Mallat and S. Zhong (1992), Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):710–732.

N. Piera-Carreté, J. Aguilar-Martin, and M. Sanchez, Mixed connectives between min and max. In *8th Inter. Symp. on Multiple Valued Logic*, 1988.

W. Silvert, Symmetric summation: A class of operations on fuzzy sets. *IEEE Transactions on Systems, Man, Cybernetics*, 9(10):657–659, 1979.

J. Waissman-Vilanova, *Construction d'un modèle comportemental pour la supervision de procédés: application à une station de traitement des eaux*. PhD thesis, LASS–CNRS, Novembre 2000.

R. Yager, On ordered weighted averaging operators in multi-criteria decision making. *IEEE Transactions on Systems, Man, Cybernetics*, 183–190, 1988.

R. Yager, On mean type aggregation. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 26(2):209–221, 1996.

R. Yager and A. Rybalov, Uninorm aggregation operators. *Fuzzy Sets and Systems*, 80(1):111–120, 1996.

R. Yager and A. Rybalov, Full reinforcement operators in aggregation techniques. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 28(6), 1998.

A. Yuille and T. Poggio (1986), Scaling theorems for zero-crossing. *IEEE Transaction for Zero-Crossing*, 8(1):15–25.

H. Zimmerman and P. Zynso, Latent connectives in human decision making. *Fuzzy Sets and Systems*, 4:37–51, 1980.

Index

F. Xhafa et al. (eds.), *Complex Intelligent Systems and Their Applications,* Springer Optimization and Its Applications 41,
DOI 10.1007/978-1-4419-1636-5, © Springer Science+Business Media, LLC 2010