Preface

Exploration, evaluation, and enhancement of modeling methods remain challenging areas within the field of information systems engineering. The benefits of model-driven development and information architectures are gaining wide recognition, not only by software companies and IT service providers, but also by industry groups such as the Object Management Group (OMG), with its promotion of Model Driven Architecture (MDA). In the interests of both agility and productivity, software engineers are increasingly exploiting the potential of tools for generating larger portions of their application code from high level models. In tune with these challenges and trends, this book provides an up-to-date coverage of central topics in information systems modeling and architectures by leading researchers in the area, including attention to both methodologies and best practices.

The contributions extend the state of the art, clarify fundamental concepts, evaluate different approaches and methodologies, and report on industrial applications of the techniques and methods discussed. With eleven chapters presented by top researchers from eight countries around the globe, the book provides a truly international perspective on the latest developments in information systems modeling, methods, and best practices. The range of topics explored is broad, while at the same time key topics are investigated at appropriate depth. This work should appeal to both practitioners and academics who wish to keep abreast of the latest developments in information systems modeling and information architecture.

Many of the topics in this book are typically not covered by traditional textbooks in analysis and design modeling. Modeling is used across a number of tasks in connection to information systems, but it is rare to see and easily compare all these uses of diagrammatical models as knowledge representation in the same book, highlighting both commonalities and differences between different kinds of modeling. The appropriateness of a modeling technique or language is often highlighted by the proponents of the technique, with little substantial evidence. This book focuses on the latest knowledge of and insights into this area gained through empirical and analytical evaluations of techniques used in practice. By providing a snapshot on the state of the art in modeling across a number of domains, this book should have lasting value as a reference in this field.

The chapters are arranged in three sections: *Conceptual Information Modeling; Modeling Approaches;* and *Modeling Frameworks, Architectures, and Applications*.

OVERVIEW OF CHAPTERS

Section I focuses on modeling of information at a truly conceptual level, where data models are presented in a way that nontechnical users of the relevant business domain can readily understand and validate. All three chapters in this section make reference to Object-Role Modeling (ORM), a fact-oriented modeling approach that represents information in terms of facts that assert that one or more objects play roles in relationships. In contrast to Entity-Relationship Modeling (ER) and the class diagramming technique within the Unified Modeling Language (UML), ORM models are attribute-free: this enables ORM models to be easily validated by verbalization and population, while at the same time conferring greater semantic stability (e.g. if one wishes in ER or UML to add an attribute of, or relationship to, an existing attribute, the latter attribute has to be remodeled as an object or relationship; in ORM the existing attribute would have been modeled as a fact type, and one may now simply add another fact type, without disrupting the original model).

Other benefits of ORM include its rich graphical constraint language, and its detailed modeling procedures for constructing conceptual schemas, validating them, and transforming them to attribute-based schemas such as ER schemas, UML class schemas, relational database schemas, or XML schemas. Chapters I and II include discussion of ER and/or UML to facilitate the application of their main ideas to the other approaches as well.

Each of the three chapters in Section I propose extensions to existing modeling approaches. Chapter I, entitled "Enriched Conceptualization of Subtyping", by Terry Halpin, suggests ways to address complex subtyping aspects that are often encountered in industrial application domains. Three classes of subtypes are distinguished: asserted, derived, and semiderived. Asserted subtypes are simply asserted or declared, whereas derived subtypes are fully defined in terms of other types that already exist. Semiderived types allow instances that are simply declared as well as instances that are derived. For subtypes that are derived or semiderived, care is needed to keep the types in sync with their derivation rules. Rigid types are distinguished from role types. An instance of a rigid type (e.g. Person) must remain in that type for the duration of its lifetime, whereas instances of role types (e.g. Student, Employee) may move into and out of those types at different times. Ways to declare rigidity are proposed for both ORM and UML, and the importance of keeping such declarations in sync with dynamic constraints such as changeability settings on association roles is highlighted. Various data model patterns are then identified to deal with modeling temporal aspects such as maintaining history of instances as they migrate from one type to another.

Chapter II, entitled "Essential, Mandatory, and Shared Parts in Conceptual Data Models", by Alessandro Artale and C. Maria Keet, focuses on formally representing life cycle semantics of part-whole relations in conceptual data models by utilizing the temporal modality. Their approach exploits the temporal conceptual data modeling language ERVT, extending it with the novel notion of status relations. This enables a precise axiomatization of the constraints for essential parts and wholes compared to mandatory parts and wholes, as well as the introduction of temporally suspended part-whole relations. A taxonomy is presented to cover both mereological and meronymic part-whole relations, and various notions such as essential parthood and essential wholehood are investigated. To facilitate usage in the conceptual stage, the authors propose a set of closed questions and a decision diagram to be used in communicating with the domain expert in order to determine just what kinds of shareability and lifetime aspects are intended for specific parthood fact types. Their work also suggests one way to provide a formal semantics for shareability based on description logics, and how to model these kinds of differences in conceptual data models.

Chapter III, entitled "Extending the ORM Conceptual Schema Language and Design Procedure with Modeling Constructs for Capturing the Domain Ontology", by Peter Bollen, extends the fact-oriented ORM approach, primarily to cater for lists of concept definitions that explain the meaning of the terms (object types) and relevant predicates (relationships) referred to in the information model. Extensions are proposed not only for the underlying metamodel but also for the modeling procedure itself, with particular attention to subtype definitions and naming conventions. The conceptual schema design procedure (CSDP) for ORM is augmented to ensure that appropriate attention is given to specifying the relevant explanations and definitions of the relevant noun and verb phrases used in expressing the model, thus improving the likelihood that users of the information system will understand the intended semantics of the model.

Section II focuses on modeling approaches, and consists of four chapters looking at broader modeling approaches including enterprise and process modeling in combination with data modeling.

In chapter IV, entitled "EKD - An Enterprise Modeling Approach to Support Creativity and Quality in Information Systems and Business Development" Janis Stirna and Anne Persson present experiences and reflections from using the EKD Enterprise Modeling method in a number of European organizations. The EKD (Enterprise Knowledge Development) modeling method is presented. The chapter then focuses on the application of EKD in practice, taking six cases as examples. The observations and lessons learned are reported concerning general aspects of Enterprise Modeling projects, the EKD modeling language, the participative modeling process, tool support, and issues of Enterprise Model quality. They also discuss a number of current and emerging trends for development of Enterprise Modeling approaches in general and for EKD in particular.

In chapter V, entitled "Integrated Requirement And Solution Modeling: An Approach Based On Enterprise Models" by Anders Carstensen, Lennart Holmberg, Kurt Sandkuhl, and Janis Stirna, the authors discuss how a particular enterprise modeling approach, namely C3S3P, has been applied in an automotive supplier company. The chapter concentrates on the phases of the C3S3P development process such as Concept Study, Scaffolding, Scoping, and Requirements Modeling. They also present the concept of task patterns which have been used for capturing, documenting and sharing best practices concerning business processes in an organization. Within this application context they have analyzed the experiences concerning stakeholder participation and task pattern development. They have also described how to derive four different categories of requirements from scenario descriptions for the task patterns and from modeling of the task patterns.

In chapter VI, entitled "Methodologies for Active Knowledge Modeling" by John Krogstie and Frank Lillehagen, the authors look deeper at AKM (Active Knowledge Modeling), which is the basis for among others the C3S3P-approach. Innovative design is the most important competitive factor for global engineering and manufacturing. Critical challenges include cutting lead times for new products, increasing stakeholder involvement, facilitating life-cycle knowledge sharing, service provisioning, and support. Current IT solutions for product lifecycle management fail to meet these challenges because they are built to perform routine information processing, rather than support agile, innovative work. AKM provides an approach, methodologies, and a platform to remedy this situation.

This chapter describes the AKM-approach applied by manufacturing industries and consultants to implement pragmatic and powerful design platforms. A collaborative product design methodology describes how teams should work together in innovative design spaces. How to configure the AKM platform to support such teams with model-configured workplaces for the different roles is described in the visual solutions development methodology. The use of this approach is illustrated through a case study in the same areas as the previous chapter, but here the resulting workplaces are presented in much more detail. The AKM-approach is also compared with related work in the enterprise modeling arena to illustrate the novelty of the approach.

The final chapter in this section, chapter VII, "Data Modeling and Functional Modeling - Examining the Preferred Order of Using UML Class Diagrams and Use Cases" by Peretz Shoval, Mark Last, and Avi Yampolsky, returns to more traditional software modeling, investigating the preferred order of two modeling techniques that are popular in UML. In the analysis phase of the information system development, the user requirements are studied, and analysis models are created. In most UML-based methodologies, the analysis activities include mainly modeling the problem domain using a class diagram, and modeling the user/functional requirements using use cases. Different development methodologies prescribe different orders of carrying out these activities, but there is no commonly agreed order for performing them.

In order to find out whether the order of analysis activities makes any difference, and which order leads to better results, a comparative controlled experiment was carried out in a laboratory environment. The subjects were asked to create two analysis models of a given system while working in two opposite orders. The main results of the experiment are that the class diagrams are of better quality when created as the first modeling task, and that analysts prefer starting the analysis by creating class diagrams first.

Section III is concerned with modeling frameworks, architectures, and applications. It essentially takes a broader perspective on the art of modeling. Individual chapters either focus on fundamental aspects of modeling methods, such as the construction of methods (Chapter VIII) or the study of heuristics and strategies to influence the quality of models (Chapter IX), or they explore modeling in specific application areas, such as service oriented software architectures (Chapter X), or web information systems (Chapter XI).

Chapter VIII, entitled "OntoFrame - An Ontological Framework for Method Engineering" by Mauri Leppänen, presents an ontological framework, called OntoFrame, that can be used as a coherent conceptual foundation for the construction, analysis and comparison of method engineering artifacts. A modular structure is used leading to multiple ontologies. For each ontology, the purpose, sub-domains and theoretical foundations are highlighted, as well as the approaches and process by which OntoFrame has been constructed. Furthermore, OntoFrame is applied to make a comparative analysis of existing conceptual artifacts.

Chapter IX, entitled "Concepts and Strategies for Quality of Modeling", by Patrick van Bommel, Stijn Hoppenbrouwers, Erik Proper, and Jeroen Roelofs, presents a process-oriented framework (QoMo) which aims to further the study of analysis and support of the processes involved in modeling. The resulting framework is strongly goal-oriented, and is expressed largely by means of formal rules. The chapter includes the discussion of a prototype implementation of the framework, which serves as an illustration and proof of concept.

Chapter X, entitled "Service Oriented Architecture - A Research Review from the Software and Applications Perspective", by John Erickson and Keng Siau, presents the basic ideas underlying service oriented architecture as used in the context of software and application architectures. Relevant historical background behind the move of software architectures towards a service oriented architectural style is presented, including antecedents such as Web Services, SOAP, and CORBA, and enabling technologies such as XML and EJB. The common components of a service-oriented software architecture, including UDDI, Application Programming Interface, Service Bus, Service Contract, Interface, Implementation, Data, and Business Logic are also presented. Finally, relevant research in four categories is presented, including implementation strategies, patterns and blueprints, tool development, standards proposals or modifications (including middleware), and ontological or meta-model development or modification.

In Chapter XI, "Designing Web Information Systems for a Framework-based Construction", by Vítor Estêvão Silva Souza, Ricardo de Almeida Falbo, and Giancarlo Guizzardi discusses a framework-based strategy for the construction of web-based information systems (WIS). More specifically, this chapter presents a design method called FrameWeb which defines a standard architecture for framework-based Web information systems and a modeling language that extends UML to build diagrams that specifically depict framework-related components.

Section IV ends the book with four chapters related to methodological aspects of modeling. In Chapter XII, "Business Process Simulation: An Alternative Modelling Technique for the Information System Development Process" by Tony Elliman, Tally Hatzakis, and Alan Serrano, another usage area for formal process models is illustrated. The article discuss the idea that even though information systems development (ISD) approaches have long advocated the use of integrated organizational views, the modeling techniques used have not been adapted accordingly and remain focused on the automated information system (IS) solution. Existing research provides evidence that business process simulation (BPS) can be used at different points in the ISD process in order to provide better integrated organizational views that aid the design of appropriate IS solutions. Despite this fact, research in this area is not extensive, which suggests that the potential of using BPS for the ISD process is not yet well understood. The article uses the findings from three case studies in order to illustrate the ways that BPS has been used at different points in the ISD process. It compares the results against IS modeling techniques, highlighting the advantages and disadvantages that BPS has over the latter. The research necessary to develop appropriate BPS tools and give guidance on their use in the ISD process is discussed

In Chapter XIII "An Agent Based Formal Approach for Modeling and Verifying Integrated Intelligent Information Systems" by Leandro Dias da Silva, Elthon Allex da Silva Oliveira, Hyggo Almeida, and Angelo Perkusich, illustrates a further usage of more formal modeling techniques. A formal agent-based approach for the modeling and verification of intelligent information systems using colored Petri nets is presented. The use of a formal method allows analysis techniques such as automatic simulation and verification, increasing the confidence on the system behavior. The agent-based modeling allows separating distribution, integration, and intelligent features of the system, improving model reuse, flexibility, and maintenance. As a case study, an intelligent information control system for parking meters price is presented.

In Chapter XIV "Design Principles for Reference Modeling: Reusing Information Models by Means of Aggregation, Specialization, Instantiation, and Analogy" by Jan vom Brocke, the focus is on generic process models, rather than concrete process models which are dealt with in an earlier chapter. With the design of reference models, an increase in the efficiency of information systems engineering is intended. This is expected to be achieved by reusing information models. Current research focuses mainly on configuration as one principle for reusing artifacts. According to this principle, all variants of a model are incorporated in the reference model, facilitating adaptations by choices. In practice, however, situations arise whereby various requirements to a model are unforeseen: either results are inappropriate or costs of design are rising strongly. This chapter introduces additional design principles aiming at giving more flexibility to both the design and application of reference models

Finally, in Chapter XV "Examining the Quality of Evaluation Frameworks and Meta-Modeling Paradigms of IS Development Methodologies", Eleni Berki gives an historical account on information systems development methodologies and associated CASE tools, and as such provides a backdrop on the framework presented in Chapter VIII. The construction and evaluation of methodologies are usually carried out by evaluation frameworks and meta-models, both considered as meta-methodologies. This chapter investigates and reviews representative meta-models and evaluation frameworks for assessing the capability of methodologies to contribute to high-quality outcomes. It presents a summary of their quality features, strengths, and weaknesses.