Preface

The field of information systems engineering includes numerous modeling methods and notations. Even with attempts to standardize (e.g., UML for object-oriented design adopted by the Object Management Group (OMG)), new modeling methods and methodologies are constantly being introduced, many of which differ only marginally from existing approaches. A systematic study of modeling methods and methodologies is needed to understand the strengths and weaknesses of each method, and the appropriate contexts and tasks where each is most suitably applied. This understanding has important consequences for the use of existing methods, evolution of existing methods, and the design of entirely new methods (method engineering) when necessary.

The purpose of this book is to disseminate the research results and best practice from researchers and practitioners interested in and working on modeling methods and methodologies. Evaluation of modeling methods remains a challenge in information systems engineering. Though the need for such studies is well recognized, there is a paucity of such research in the literature. There is a clear need for innovative, effective, and efficient techniques for such evaluation. The EMMSAD (Evaluating Modeling Methods for System Analysis and Design) workshops have since 1996 been a meeting ground specifically to attack this problem area. The book is based on extended versions of the best papers submitted to the EMMSAD workshops in 2001, 2002, and 2003. The EMMSAD workshop series has received wide acceptance and support from the modeling community. Each year, we have received submissions from all over the globe, making EMMSAD a truly international workshop, with 30 to 50 participants attending each year, both from research and industry

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 the underlying principles of, e.g., enterprise modeling, process modeling, requirements modeling, and design modeling. What specifically distinguishes this book is that it looks across these various research domains, to provide an overview of existing approaches and best practices in these conceptually closely related fields.

The chapters in this book are not traditionally covered by traditional textbooks in analysis and design modeling. Textbooks on more advanced modeling-oriented topics are mainly focusing on the favorite techniques of the authors, and they often present little coverage on formal evaluation of different modeling techniques. Some books present briefly a number of techniques for a specific domain, but seldom look across domains to see how modeling as a generic technique can be generally applied, and which aspects need to be specifically tailored to the modeling task at hand.

The chapters are arranged within three sections: *General Techniques for Comparing and Adapting Modeling Methods; Goal, Requirements, and Process Modeling;* and *Data, Ontology, and Component Modeling.*

OVERVIEW OF CHAPTERS

Section I focuses on general ways of comparing and extending modeling approaches, including the use of meta-modeling, assessment of quality of models and modeling languages, and method engineering. In the chapter "Analyzing and Comparing Ontologies with Meta-Models," by Islay Davies, Peter Green, Simon Milton, and Michael Rosemann, the authors propose the use of meta-models for analyzing, comparing, and engineering ontologies. High-level ontologies provide a model of reality and are of increasing popularity for the evaluation of modeling methods. The chapter discusses a methodology using extracts of meta-models for two well-known ontologies that had been used previously in systems analysis and design research. The approach provides a theoretical analysis technique for evaluating these ontologies according to their equivalence, depth of structure, and comprehensiveness of scope.

The focus in Chapter II, "Two Meta-Models for Object-Role Modeling," by Dave Cuyler and Terry Halpin, is also the use of meta-modeling techniques. Although object-role modeling (ORM) has been used for three decades and now has industrial modeling tool support, it has no official, standard meta-model. Because of its extensive capability for expressing business rules, ORM is currently being considered as a possible standard for business rules expression within the object management group (OMG), and for use in ontology standards. To support these initiatives, and the interchange of ORM model data between different software tools, this chapter discusses recent research by the authors to pave the way for a standard ORM meta-model that employs a standard interchange format.

Bart-Jan Hommes similarly applies meta-modeling techniques in Chapter III, "Evaluating Conceptual Coherence in Multi-Modeling Techniques." A meta-model can serve as a basis for quantitative evaluation of methods and techniques. By means of a number of formal metrics based on the meta-model, a quantitative evaluation of methods and techniques becomes possible. Existing meta-modeling languages and measurement schemes do not allow the explicit

modeling of so-called multi-modeling techniques. Multi-modeling techniques are techniques that offer a coherent set of aspect modeling techniques to model different aspects of a certain phenomenon. As a consequence, existing approaches lack metrics to quantitatively assess aspects that are particular to multi-modeling techniques. In this chapter, a modeling language for modeling multi-modeling techniques is proposed, as well as metrics for evaluating the coherent set of aspect modeling techniques that constitute the multi-modeling technique.

In Chapter IV, "Assessing Enterprise Modeling Languages Using a Generic Quality Framework," by John Krogstie and Sofie de Flon Arnesen, an even broader framework of criteria for the goodness of a modeling language is presented and used for the evaluation in a practical setting. The users of the framework have recently wanted to standardize an enterprise modeling language for process modeling for sense-making and communication. To perform the evaluation, a generic framework for assessing the quality of models and modeling languages was specialized to the needs of the company. The work illustrates the practical utility of the overall framework, where language quality features are looked upon as means to enable the creation of models of high quality. It also illustrates the need for specializing this kind of general framework based on the requirements of the specific organization.

The last chapter in Section I is titled, "An Approach for Evolution-Driven Method Engineering," by Jolita Ralyté, Colette Rolland, and Mohamed Ben Ayed. The chapter considers the development of new methods, particularly the evolutionary perspective of method engineering. It presents an approach for method engineering supporting evolution of an existing method, model, or metamodel into a new one satisfying a different engineering objective. This approach proposes several different strategies to evolve from the initial methodology to a new one and provides guidelines supporting these strategies. The approach has been evaluated in the research project around the Lyee methodology.

Section II looks at methods and methodologies for behavioral modeling in a broad sense, including both goal-oriented modeling, requirements modeling, and process modeling

In the field of requirements engineering, goal modeling approaches have received much attention in recent years by researchers and practitioners alike. In Chapter VI, "Goal Modeling in Requirements Engineering: Analysis and Critique of Current Methods," Evangelia Kavakli and Pericles Loucopoulos identify the uses of these approaches in different contexts of requirements analysis phases. It provides an analysis of goal modeling approaches in a systematic and consistent manner. The aim of this analysis is to understand the best fit for purpose of different goal modeling approaches and to highlight open issues that provide a foundation for further research in this important area of requirements engineering methodology.

Chapter VII, "An Empirical Investigation of Requirements Specification Languages: Detecting Defects While Formalizing Requirements," by Erik Kamsties, Antje von Knethen, Jan Philipps, and Bernhard Schätz, presents an empirical study of requirements specification languages, in which two research questions are addressed: Which types of defects are detected by a requirements engineer during formalization? Which types of defects go undetected and what happens to those types in a formal specification? The results suggest looking explicitly for ambiguities during formalization, because they are less frequently detected than other types of defects. If they are detected, they require immediate clarification by the requirements author.

Automated support for the requirements engineering process is a recognized research area. However, the mainstream practice still relies on word processors and drawing tools rather than the requirements engineering tools (RETs). The aim of Chapter VIII, "Validating an Evaluation Framework for Requirements Engineering Tools," by Raimundas Matulevičius, is to validate an evaluation framework for RETs. The validation process concerns an RET acquisition process for concrete organizational needs. An observation of maintaining requirements specification shows the important organizational and environmental characteristics for a proper automated support of RE process. The contribution of this work is twofold: first, the validation of the evaluation framework for RETs according to environmental needs in a specific environment, and second, the identification of environmental needs, which emerge from the requirements specification maintenance process.

Process modeling is often used for analysis purposes. In Chapter IX, "A Comparison of the FOOM and OPM Methodologies for User Comprehension of Analysis Specifications," by Judith Kabeli and Peretz Shoval, two approaches to integrated process and OO-modeling is compared. FOOM (functional and object-oriented methodology) and OPM (object-processes methodology) are methodologies used for analyzing and designing information systems. In this study, the authors compare FOOM and OPM from the point of view of both user comprehension of analysis specifications and user preference of specifications. The comparison is based on a controlled experiment that measured: (a) comprehension of the analysis specifications, which includes both structural and behavioral aspects of the system; (b) the time it takes to complete the task of specification comprehension; and (c) the user's preference of models.

The last chapter in Section II follows up on process modeling in practice. In Chapter X, "Participatory Development of Enterprise Process Models," Reidar Gjersvik, John Krogstie, and Asbjørn Følstad present practical experience from using a technique called Modeling Conference, a method for participatory construction and development of enterprise process models. The Modeling Conference method focuses on broad participation from all actors in the organization, is grounded in a social constructivist perspective, and has its theoretical basis in the method of search conferences and process modeling. In an engi-

neering consultancy firm, the Modeling Conference method has been used to develop process models for main common work tasks that have been implemented on an intranet. Independent evaluations show that participation through the Modeling Conference led to significantly more ownership to the process models, and that the actors developed new collective knowledge.

Section III looks at techniques for structural modeling, both on an analysis and design level. This includes looking at various data and class-oriented modeling approaches, modeling of ontologies, and modeling of system components.

Discovering a set of domain classes during object-oriented analysis is intellectually challenging and time consuming for novice analyzers. In Chapter XI, "A Taxonomic Class Modeling Methodology for Object-Oriented Analysis," Il-Yeol Song, Kurt Yano, Juan Trujillo, and Sergio Luján-Mora present a taxonomic class modeling (TCM) methodology that can be used for object-oriented analysis in business applications that helps to discover three types of classes: (1) classes represented by nouns in the requirement specification, (2) classes whose concepts were represented by verb phrases, and (3) hidden classes that were not explicitly stated in the requirement specification. The framework integrates the noun analysis method, class categories, English sentence structures, checklists, and other heuristic rules for modeling.

Data models of realistic domains often become large and difficult to work with. In Chapter XII, "Comprehension of Hierarchical ER Diagrams Compared to Flat ER Diagrams," by Revital Danoch, Peretz Shoval, and Mira Balabaan, HERD — a semi-algorithmic method for creating hierarchical ER diagrams from bottom up — is presented. The method is based on packaging operations that are applied in several steps on a given flat ER diagram. The result is a hierarchy of simple and interrelated diagrams (namely ER structures) with external relationships to other such diagrams. The chapter describes the application of HERD method using an example from a hospital domain, and an experiment in which the authors compare the comprehension of HERD diagrams with flat ER diagrams.

To ensure that a software system accurately reflects the business domain that it models, the system needs to enforce the business rules (constraints and derivation rules) that apply to that domain. From a conceptual modeling perspective, many application domains involve constraints over one or more conceptual schema paths that include one or more conceptual joins (where the same conceptual object plays roles in two relationships). In Chapter XIII, "Constraints on Conceptual Join Paths," Terry Halpin contrasts how these join constraints are catered for in object-role modeling (ORM), the unified modeling language (UML), the object-oriented systems model (OSM), and some popular versions of entity-relationship (ER) modeling (ER). Three main problems for rich support for join constraints are identified: disambiguation of schema paths, disambiguation of join types, and mapping of join constraints to implementation code. To address these problems, some notational, meta-model, and mapping extensions are proposed.

Another type of representing structural aspects are the modeling of ontologies. In Chapter XIV, "Using a Semiotic Framework for a Comparative Study of Ontology Languages and Tools," Xiaomeng Su and Lars Ilebrekke survey and compare different ontology languages and tools by the aid of an evaluation framework (the same framework that was used in Chapter IV). An ontology must be of high quality to enable actors to reach a common understanding of the domain at hand. The notion of "quality" in the context of ontology is discussed, and means to achieve high-quality ontologies are listed. The different quality aspects and means to improve them formulate the template for the comparisons of ontology languages and tools, which are two of the major factors that affect the quality of ontologies.

Component-based development (CBD) has received a lot of attention in software engineering literature over the last few years. Awareness has been raised that CBD is the way to go in software development, especially in the domain of e-business, where the benefits of reusing components, i.e., faster time-to-market and quality, are essential. In Chapter XV, "A Service-Oriented Component Modeling Approach," Zoran Stojanovic, Ajantha Dahanayake, and Henk Sol present a service-oriented component modeling approach focused on the concepts of component and service as the main modeling and design artifacts.

In the final chapter, "Evaluation of Component-Based Development Methods," Nicky Boertien, Maarten W.A. Steen, and Henk Jonkers present an evaluation of five popular methods for component-based development — including Catalysis, the Rational Unified Process, and Select Perspective — on their maturity and fitness-for-use in the context of e-business engineering. The evaluation is done based on their own reference framework for e-business development and a list of objective criteria. The methods each emphasize certain aspects of CBD, but as yet none of them offers a complete solution.

The above collection of chapters provides a good mix of applications of modeling techniques across a number of knowledge representation problems within information system analysis and design. The goodness of a modeling technique or language is often highlighted by the proponents of the technique with little substantial evidence. In addition to give an overview of the current state of the art and state of practice within information modeling methods and methodologies, the book focuses on our knowledge of this area through empirical and analytical evaluations of techniques used in practice.