Domain-Specific Conceptual Modeling
Domain-Specific Conceptual Modeling

Concepts, Methods and ADOxx Tools
A Note from the Editors of the Previous Volume “Domain-Specific Conceptual Modeling”

The predecessor volume *Domain-Specific Conceptual Modeling: Concepts, Methods and Tools* (henceforth called “volume I”) published in 2016 represented at that time a novelty in the field of conceptual modeling. It collects the work of different international research groups, each focusing on a domain-specific modeling method, accompanied by proof of concept within its tool implementation based on ADOxx.

A detailed overview of the domain-specific conceptual modeling methods and tools, developed by members active in the OMiLAB Community of Practice showcased in volume I, is given in the chapter “The Purpose-Specificity Framework for Domain-Specific Conceptual Modelling” by Robert A. Buchmann.

The success and positive reception of the first volume among researchers, practitioners, as well as students eager to expand and strengthen their knowledge in conceptual modeling led to our suggestion to continue with a further publication. We are thankful to the scientific community that volume I motivated new researchers and that the new team of editors took our advice and presented these results in this book, showing that this field has continuity and plays a key role, especially in the digital age.

Guided by domain experts, the work of recently established research groups will continue to have an impact and benefit academia and industry, interested students, as well as members of the conceptual modeling community to embrace digital transformation initiatives and projects.

As editors of volume I, we are confident that the new volume will be received by the conceptual modeling community with at least the same outstanding response!

October 2021

Dimitris Karagiannis
Heinrich C. Mayr
John Mylopoulos
Preface

Following the goal of the previous volume, *Domain-Specific Conceptual Modeling: Concepts, Methods and Tools*, to “increase the visibility of domain-specific modelling,” in this volume, the significance of the topic is once more demonstrated through new research and development approaches that are manifested in each of the chapters. These include novel modeling methods and tools that emphasize the recent results accomplished and the adequacy to assess specific aspects of a domain.

This successor volume, *Domain-Specific Conceptual Modeling: Concepts, Methods, and ADOxx Tools*, highlights again the work of researchers who have designed and deployed domain-specific modeling methods and tools in the context of the OMiLAB (www.omilab.org) Community of Practice.

Each chapter offers detailed instructions on how to build models in a particular domain, such as product-service engineering, enterprise engineering, digital business ecosystems, and enterprise modeling and capability management. Furthermore, they emphasize possible future developments and research directions in an open manner. All these achievements are enriched with case studies, related information, and tool implementation. The tools are based on the ADOxx metamodelling platform and are provided free of charge via OMiLAB.

The accomplishments are embedded in the OMiLAB approach, more specifically introducing the Agile Modelling Method Engineering (AMME) approach, the ADOxx (www.adoxx.org) platform for experimental realization, the Digital Innovation Environment (DiEn) powered by OMiLAB, and a synopsis on the results of the previous volume in the context of new developments.

We are confident that the theoretical foundation and collection of domain-specific modeling methods and tools presented in this volume will benefit experts and practitioners from academia and industry alike, including members of the conceptual modeling community as well as researchers, lecturers, and students.

A large scientific community was involved in creating this volume, and we would like to express our gratitude to each and every one for their contribution: First of all, we thank all the authors who submitted their work and provided their expertise in the field, and reviewers for their constructive and helpful feedback.
We are also thankful for the support received from the team at Springer led by Ralf Gerstner in the publication of the book!

We highly appreciate the efforts of all those involved!

OMiLAB NPO envisions an active global community for conceptual modeling that benefits from open artifacts. To this end, it acts as a facilitator to the development of scientific methods and technologies for all those who value models. In addition, it is a platform where participants can bring in ideas related to modeling and engage in the exploration process. Get in touch with us via info@omilab.org to get involved in the OMiLAB Community of Practice!

Fig. 1  ADOxx-based tools from volume I and volume II

Vienna, Austria          Dimitris Karagiannis
Jeonju, Korea (Republic of) Moonkun Lee
Olten, Switzerland       Knut Hinkelmann
Berlin, Germany          Wilfrid Utz
October 2021
## Contents

### Part I  Background

1  Conceptual Modelling Methods: The AMME Agile Engineering Approach ...................................................... 3  
   Dimitris Karagiannis

2  Development of Conceptual Models and Realization of Modelling Tools Within the ADOxx Meta-Modelling Environment: A Living Paper ............................................. 23  
   The OMiLAB Community

3  Challenging Digital Innovation Through the OMiLAB Community of Practice .................................................... 41  
   Iulia Vaidian, Arkadiusz Jurczuk, Zbigniew Misiak, Michael Neidow, Martin Petry, and Martin Nemetz

### Part II  Previous Volume: Synopsis

4  The Purpose-Specificity Framework for Domain-Specific Conceptual Modeling ...................................................... 67  
   Robert Andrei Buchmann

### Part III  Enterprise Management

5  Enterprise Modeling with 4EM: Perspectives and Method ............ 95  
   Birger Lantow, Kurt Sandkuhl, and Janis Stirna

6  PGA 2.0: A Modeling Technique for the Alignment of the Organizational Strategy and Processes .............................. 121  
   Ben Roelens

7  The LiteStrat Modelling Method: Towards the Alignment of Strategy and Code .................................................. 141  
   Oscar Pastor, Rene Noel, Ignacio Panach, and Marcela Ruiz
8 itsVALUE: Modelling and Analysing Value Streams for IT Services ................................................................. 161
Henning Dirk Richter, Birger Lantow, and Thomas Pröpper

Part IV Enterprise Information Systems

9 Enterprise Construction Modeling Method ........................................ 187
Marné De Vries and Jan L. G. Dietz

10 Tool Support for Fractal Enterprise Modeling .......................... 205
Ilia Bider, Erik Perjons, and Victoria Klyukina

11 The Integration of Risk Aspects into Business Process Management: The e-BPRIM Modeling Method ......................... 231
Elyes Lamine, Rafika Thabet, Amadou Sienou, and Hervé Pingaud

12 Modeling the Phenomenon of Capability Change: The KYKLOS Method .......................................................... 265
Georgios Koutsopoulos, Martin Henkel, and Janis Stirna

Zacharenia Garofalaki, Dimitrios Kallergis, and Christos Douligeris

Part V Business Ecosystems and Services

14 A Modeling Tool for Exploring Business Ecosystems in a (Pre-)conceptual Phase ..................................................... 315
Florian Schierlinger-Brandmayr, Birgit Moesl, Philipp Url, Wolfgang Vorraber, and Siegfried Vössner

15 A Capability-Based Method for Modeling Resilient Data Ecosystems ................................................................. 339
Jānis Grabis, Līva Deksne, Evita Roponena, and Janis Stirna

16 Space of Services Method (SoS) ........................................... 365
Vjeran Strahonja, Zlatko Stapić, and Martina Tomićić Furjan

17 Design and Engineering of Product-Service Systems (PSS): The SEEM Methodology and Modeling Toolkit ......................... 385
Fabiana Pirola, Giuditta Pezzotta, Danial Mohammadi Amlashi, and Sergio Cavalieri

Part VI Knowledge Engineering

18 Model-Based Guide Toward Digitization in Digital Business Ecosystems ................................................................. 411
Anna Sumereeder and Tor Dokken
19 Generating ROS Codes from User-Level Workflow in PRINTEPS
   Takeshi Morita and Takahira Yamaguchi

20 ECAVI: An Assistant for Reasoning About Actions and
   Change with the Event Calculus
   Nena Basina, Theodore Patkos, and Dimitris Plexousakis

Part VII Technology Enhanced Education

21 Tree Diagrams and Unit Squares 4.0: Digitizing Stochastic
   Classes with the Didactic Modeling Tool PROViS
   Victoria Döller and Stefan Götz

22 Improving Student Mobility Through Automated Mapping
   of Similar Courses
   Martina Tomičič Furjan, Bogdan Okreša Djuric, and
   Tomislav Peharda

Part VIII Digital Humanities

23 Aggregation and Curation of Historical Archive Information
   Panos Constantopoulos, Vicky Dritsou, Maria Ilvanidou, and
   Alexandra Chroni

Part IX Modelling Method Conceptualization

24 Conceptualization of Modelling Methods in the Context
   of Categorical Mechanisms
   Daniel-Cristian Crăciunean and Daniel Volovici

25 Conceptualizing Design Thinking Artefacts: The
   Scene2Model Storyboard Approach
   Christian Muck and Silke Palkovits-Rauter

26 An Approach to the Information System Conceptual
   Modeling Based on the Form Types
   Ivan Luković, Milan Čeliković, Slavica Kordić, and Marko Vještica

Part X Conceptual Modelling Language Extension

27 BPMN4MoPla: Mobility Planning Based on Business
   Decision-Making
   Emanuele Laurenzi, Oliver Ruggli, and Alta van der Merwe

28 BPMN Extension for Multi-Protocol Data Orchestration
   Andrei Chis and Ana-Maria Ghiran
About the Editors

**Dimitris Karagiannis** holds a full professor position for Business Informatics at the University of Vienna since 1993, leading the Research Group Knowledge Engineering (www.dke.univie.ac.at). He received his PhD degree from the Technical University of Berlin in 1987. The same year he joined the Research Institute for Application-oriented Knowledge Processing in Ulm as division head for “Enterprise Information Systems.” Prof. Karagiannis holds an honorary professorship from the Babes-Bolyai University in Cluj-Napoca, Romania. His research interests include metamodelling, knowledge engineering, business process management, enterprise architecture management, and artificial intelligence. The industrial application of his metamodelling research was demonstrated within the BOC Group (www.boc-group.com), a European software and consulting company founded in 1995. In parallel, scientific applications of his research are applied in the Open Models Laboratory—OMiLAB, http://www.omilab.org, an open collaborative environment for modeling method engineering, which he has established and is currently leading, located in Berlin.

**Moonkun Lee** holds a full professor position for Computer Science and Engineering at Jeonbuk National University (JBNU) in the Republic of Korea since 1996. He received a Bachelor’s degree in Computer Science from Pennsylvania State University in the USA in 1989, and a Master’s and Ph.D. degrees in Computer and Information Science from the University of Pennsylvania in the USA in 1992 and 1995, respectively. He worked at CCCC in the USA as Computer Scientist from 1992 to 1996 and developed SRE (Software Re/reverse-engineering Environment) applied to the modernization of legacy OS and SW of NSWC in US Navy to Ada. His main research interests are SW round-trip engineering, distributed real-time systems, formal methods, ontology, behavior engineering, etc. He published a number of research papers in journals and conferences, related to the family of *delta-Calculus*, including *dT* and *dTP-Calculus*. He also developed a mathematical structure called *n:2-Lattice* to define the graphical notion of *behavior ontology*. Further, he published several books in Korean, among which most noticeable are *Formal Methods* (2017, JBNU Press) and *Theory of Multi-Paradigm Programming*.
Languages (2021, JBNU Press). For the implementation of the calculus and the ontology, he developed the SAVE and the PRISM tools on the ADOxx metamodeling platform. In industrial applications, one of the most interesting research with SAVE was to prove the integrity of a new engine from a research center of Hyundai Motor Company in 2018 by generating more than 1 million safe cases of the engine states without any deadlock. Currently, he focuses on the specification, analysis, verification, and implementation of the CPS (Cyber-Physical Systems) for Smart City and Industry in SAVE on the ADOxx and the OLIVE with IoT devices in terms of probabilistic models for non-deterministic behavior of the AI applications.

Knut Hinkelmann is Professor for Information Systems and Head of the Master of Science in Business Information Systems at the FHNW University of Applied Sciences and Arts Northwestern Switzerland, where he also is head of the research group Intelligent Information Systems. He is Managing Director of OMiLAB NPO since 2020. Since 2015, he has been a Research Associate at the University of Pretoria (South Africa) and since 2017 he is an Adjunct Professor at the University of Camerino (Italy). His research interests are Enterprise Modeling and Artificial Intelligence. He has served as program chair of several international conferences and has published more than 130 publications. In 1988, he obtained a diploma in Computer Science from the University of Kaiserslautern and a Ph.D. in Natural Sciences from the Computer Science Department of the same university in 1995. From 1988 to 1990, he was a researcher at the Research Institute for Applied Knowledge Processing FAW in Ulm. From 1990 until 1998, he was a researcher and later Head of the Knowledge Management research group at the German Research Center for Artificial Intelligence, DFKI. From 1998 until 2000, he worked as product manager for Insiders Information Management GmbH. He was CEO of the KIBG GmbH from 1996 until 1998, and from 2006 until 2012 he was Scientific Advisor of STEAG & Partner AG.

Wilfrid Utz is one of the managing directors of OMiLAB NPO, the non-profit organization headquartered in Berlin supporting the conceptual modeling community organized around emerging topics with respect to domain-specific conceptual modeling. Wilfrid completed his PhD thesis in 2020 at the University of Vienna in the field of metamodel design and knowledge representation using conceptual structures. He has been involved in numerous international research and innovation projects and gained experience in the field of modeling method conceptualization, design, and implementation of modeling tools using the open ADOxx metamodeling platform (www.adoxx.org).
Part I
Background
Chapter 1
Conceptual Modelling Methods: The AMME Agile Engineering Approach

Dimitris Karagiannis

Abstract Current research in fields such as Business Process Management, Enterprise Architecture Management, Knowledge Management and Software Engineering raises a wide diversity of requirements for Conceptual Modelling, typically satisfied by Design Science artefacts such as modelling methods. When employed in the context of an Agile Enterprise, an underlying requirement for Conceptual Modelling agility emerges—manifested not only on model content level but also on modelling method level. Depending on the questions that must be answered and the systems that must be supported with modelling means, the need for agility may stem from the degree of domain-specificity, from gradual understanding of modelling possibilities, from evolving model-driven systems, etc. The hereby proposed Agile Modelling Method Engineering (AMME) approach thus becomes necessary to extend the traditional perspective of “modelling through standards”; consequently, the benefits of repeatability and wide adoption are traded for responsiveness to dynamic needs identified within an Agile Enterprise.

Keywords Agile Modelling Method Engineering · Metamodelling · Conceptual Modelling · Knowledge Management · Agile Enterprise
1.1 Introduction

A diffuse notion of Agile Enterprise has emerged in the literature, as an umbrella term covering new challenges derived from increasingly dynamic needs that must be addressed by evolving and responsive enterprise functions. Agility is generally defined in relation with change: “comprehensive response to the business challenges of profiting from rapidly changing [...] global markets” [1]; “[the agile enterprise is] built on policies and processes that facilitate speed and change ...” [2]. The requirement for agility is raised both from a technical perspective (e.g., considering the high dynamics of paradigms such as Industry 4.0 [3] or the Internet of Things [4]) and from a managerial perspective (e.g., Agile Manufacturing [5], Agile Knowledge Management [6]).

Consequently, specific challenges are also emerging for the paradigm of Conceptual Modelling, considering the evolving nature of modelling needs with respect to various functions within an Agile Enterprise. Modelling requirements reclaim flexibility and agility not only for model contents (already addressed in software engineering by the Agile Modelling approach [7]), but also for the adopted modelling language, modelling software and the encompassing modelling method (the relation between these will be established in Sect. 1.3). A methodology and a new modelling paradigm are therefore necessary to address the domain-specificity of the system to be modelled, as well as the evolution of case-specific modelling requirements, for which standards may be insufficiently flexible.

The fields of Business Process Management (BPM), Enterprise Architecture Management (EAM), Model-driven Software Engineering (MDSE) and Knowledge Management (KM)—selected here as representative practices within an Agile Enterprise—have traditionally relied on conceptual modelling standards for the benefits of repeatability and reusability across domains. However, in the pursuit of the “Agile Enterprise” status, the transformative effect of the Agile Manifesto [8] (originally advocated in the context of MDSE) must also be considered for the practice of modelling method engineering in general. Regardless whether a modelling method is subordinated to an Information Systems engineering method or to various management and decision-making practices, multiple factors may generate fluctuating requirements that should be addressed by agile conceptualisation methodologies.

In support of this underlying need for agility, the framework of Agile Modelling Method Engineering (AMME, initially outlined in [9]) is hereby proposed. In addition, a community-oriented research environment—the Open Models Initiative Laboratory (OMiLAB [10])—, where the framework has been applied in several projects, will be described. Two projects will be highlighted to emphasise the applicability of AMME: (1) a research-oriented project addressing KM and EAM concerns (the ComVantage method [11] and tool [12]) and (2) an educational project for teaching MDSE and BPM topics (the FCML method [13] deployed as the BEE-UP tool [14]).
The remainder of the paper is organised as follows: Section 1.2 will overview the key motivating factors for modelling method agility, illustrated for the selected fields of BPM, EAM, KM and MDSE. Section 1.3 will describe the key facets of modelling method agility and the AMME framework. Section 1.4 will share experience and results with applying AMME in projects that have been managed within the OMiLAB environment. The paper ends with a summary and an outlook to future challenges for further consolidating AMME as a method engineering paradigm.

1.2 Conceptual Modelling for the Agile Enterprise: A Selection

A selection of fields that are highly relevant for an Agile Enterprise are discussed here as application areas for Conceptual Modelling, in order to motivate the relevance of agile modelling methods with respect to their dynamic needs.

**Conceptual Modelling for BPM** is typically associated with popular languages such as BPMN [15], EPC [16], UML activity diagrams [17] or various flowcharting predecessors that have emerged along the history of Enterprise Modelling. Petri Nets [18] became a popular choice for formalisation concerns [19] (rather than a stakeholder-oriented language). Figure 1.1 suggests a semantic spectrum that may be subject to evolving modelling requirements: (1) at the “generic” end of the spectrum, UML activity diagrams may be used to describe any type of workflow (business processes, algorithms etc.), their domain-specificity being commonly left to human interpretation; (2) BPMN diagrams narrow down semantics by fixing several concept specialisations (e.g., manual task, automated task); (3) at the right end of the spectrum, AMME was employed to semantically enrich the Task concept with a “concept schema” comprising machine-readable properties (e.g., different types of times, costs) that are relevant for decision-making or for simulation mechanisms required by stakeholders. Other BPM scenarios benefitting from AMME include (1) notational heterogeneity—i.e., when multiple business process notations co-exist and a semantic integration is required [20]; (2) the extension of business process models with conceptual patterns for semantic evaluations [21]; (3) the customisation of processes for the specificity of product-service systems [22].

**Conceptual Modelling for EAM** also benefits from various standards—e.g., Archimate [23], IDEF [24], or frameworks having a rather ontological scope without necessarily imposing diagrammatic designs (e.g., Zachman’s framework [25]). Typically, EAM employs multi-perspective methods with viewpoints that can be instantiated in various modelling views (see also ARIS [16, 26], BEN [27, 28] and MEMO [29, 30] where the multi-perspective nature is emphasised). These may also be subjected to modelling requirements that reclaim a gradual domain-specificity in the language or the method itself (as shown in the case of BPM); another common requirement is for semantic enablers to support decision-making.
mechanisms (commonly pertaining to business-IT alignment challenges). For this, a minimal necessity is consistency management across viewpoints. Figure 1.2 shows a multi-view modelling tool for the SOM enterprise modelling method [31], where changes in one model are required to propagate in the others according to semantic overlapping and dependencies—AMME is called to extend the method with consistency-preservation mechanisms that are tightly coupled with the language vocabulary (different approaches to multi-view modelling may also be consulted in [32–35]).

**Conceptual Modelling for KM** is less reliant on standard modelling languages, at least when the focus is on management practices, rather than KM systems or knowledge representation. The KM community is particularly concerned with knowledge processes such as acquisition, externalisation and learning (also in the focus of an Agile KM approach) and several key processes have been systematised in Nonaka’s seminal cycle of knowledge conversion [37]. Figure 1.3 shows how this cycle may be extended when employing Conceptual Modelling methods for knowledge representation. The following phases are hereby proposed: (1) *human-human socialisation* corresponds to Nonaka’s traditional “socialisation” phase; (2) *externalisation in raw form* corresponds to Nonaka’s traditional “externalisation” phase, if knowledge is captured in semi-structured content (to be managed with content management system); (3) *externalisation in diagrammatic-form* is enabled by modelling methods that enable knowledge acquisition through diagrammatic means (e.g., work procedures described in models rather than natural language); (4) *combination* corresponds to Nonaka’s traditional “combination” phase, with additional opportunities for combining diagrammatic knowledge representations; (5) *internalisation at machine-level* is enabled if the models are further exposed as a knowledge base to model-driven systems; (6) *machine-to-human socialisation*
would (potentially) be a socialisation variant where the “shared doing” involves a human and a knowledge-driven system (e.g., robots). The challenge of AMME in this context is to facilitate the knowledge acquisition with modelling means and tool support that are adequate to the semantics deemed relevant for KM practices and systems. Other approaches to the interplay between KM and modelling practices, based on business process modelling as a facilitator, have been overviewed in [38].

**Conceptual Modelling for MDSE** typically relies on modelling languages tailored for software design and development—e.g., UML [17], ER [39]. A popular underlying ambition is that of code generation, a task that depends on a fixed and well-defined semantic space (hence an invariant modelling language amenable to standardisation). Agile Modelling [7] is employed as a matter of quickly adapting model contents and procedures rather than the governing language. AMME becomes relevant here by raising the level of abstraction for MDSE agility, as it allows the propagation of change requests to the language semantics and further to modelling functionality. This, of course, limits the “modelling is programming” [40] possibilities (e.g., code generation); instead, AMME is motivated by a “modelling is knowledge representation” perspective, with a model base that drives “model-aware” run-time systems that are parameterised with knowledge items (rather than generated). Figure 1.4 suggests an approach proposed by the ComVantage project, where app orchestrations are derived from app requirements captured in diagrammatic form, indicating the precedence of mobile app support along a business process [42].

BPM, EAM, KM and MDSE are several fields that, under the hereby discussed assumptions and driven by project-based requirements, have motivated the emergence of AMME. The literature reports on several other approaches related
to AMME in certain aspects, however typically subordinated to MSDE goals and focusing on the domain and case specificity aspect rather than the agility of the modelling method artefact—e.g., the notion of “Situational Methods” for Information Systems Engineering [43, 44], the Domain-specific Modelling Language design methodology [45], extensibility mechanisms for standard languages [46]. Metamodelling environments such as [47–49] have significantly contributed to increasing the productivity of modelling tool implementation, thus providing candidate environments for the rapid prototyping support needed during an AMME deployment.

1.3 The AMME Framework

The notion of Agile Enterprise opens a wider scope for agility than the one advocated in agile software development and its conceptual dynamics must be captured in adequate conceptualisation and engineering processes. A classification of change drivers for an Agile Enterprise is proposed here, as illustrated in Fig. 1.5:
Fig. 1.4 Models for “model-aware information systems” (adapted from [41])

- Changes in the business model and value proposition—e.g., shifting the value proposition towards the servitisation of existing products, a deeper specialization of products reclaiming new domain-specific properties in design decisions;
- Changes in management strategy—e.g., shifting between different KM approaches or process improvement methods, reclaiming the inclusion of new properties in key performance indicators;
- Changes in support technology and infrastructure—e.g., migration to a bring-your-own-device strategy;
- Digitisation of assets—e.g., migration to new technological paradigms (Internet of Things, Industry 4.0);
- Changes in the business context—e.g., market changes, reconfigurations of virtual enterprises;
- Self-initiated changes—e.g., pro-active process re-engineering, adoption of a capability-driven Enterprise Architecture [50];
- Normative changes—e.g., changes pertaining to legal or certification compliance, evolution of already adopted standards.
- Changes in the social eco-system—e.g., changes in user behaviour, in interactions between users or between users and systems.

The enterprise performance, from an information and communication technology viewpoint, is primarily supported by (1) Enterprise Information Systems (EIS) employed at run-time (e.g., for enacting business processes and managing resources) and (2) an Enterprise Architecture (EA) supporting design-time decisions (e.g., business-IT alignment). Conceptual Modelling practices traditionally support both facets: they can enable the deployment of model-based EIS as part of some IS engineering method; they can also enable the accumulation of a Knowledge Base in conceptual model form. In both cases, modelling activities must be supported by a modelling method and adequate tooling—i.e., modelling software that supports
communication, sense-making, the accumulation of knowledge assets or analytical system designs, etc.

For this purpose, various model-based management and engineering practices typically employ available standards or well-established languages and methods. These bring inherent governance benefits (e.g., repeatability, compatibility)—however, the general assumption for adopting such methods is that modelling requirements are fixed and a standards-oriented modelling culture can be uniformly established within the enterprise and for its application domain. The hereby discussed AMME framework is motivated by the assumption that modelling requirements evolve due to one or more of several factors:

- users become gradually familiar with modelling possibilities;
- richer semantics become necessary, either for design-time (e.g., decision-support) or run-time use cases (e.g., model-driven systems);
- stakeholders gain gradual insight and common understanding of the application domain, of the properties that are relevant to the model abstractions.

Under these assumptions, the Agile Modelling Method Engineering (AMME) approach (providing several qualities suggested in Fig. 1.5) becomes necessary and the benefits of standards may be traded for other benefits—e.g., gradual domain-
specific enrichment of the modelling language, in-house evolution of model-aware systems.

Agility, as understood by AMME from an internal perspective, has two basic manifestations: (1) artefact agility is enabled by the decomposition of a modelling method into building blocks that define the backlog items to be managed through agile engineering efforts; and (2) methodological agility manifests in the engineering process itself, taking the form of an incremental and iterative spiralling development.

Artefact agility is enabled by the definition of a modelling method. The artefact created by AMME was originally defined in [51] in terms of its building blocks (Fig. 1.6):

• A modelling language further decomposed in notation (graphical symbols corresponding to the language concepts), syntax (the language grammar and associated syntactic constraints) and semantics (language vocabulary, machine-readable properties of each concept, associated semantic constraints);
• Mechanisms and algorithms cover the model-based functionality to be made available in a modelling tool—either generic (applicable to models of any type), specific (applicable only to models of a specific type) or hybrid (applicable to a limited set of model types that fulfil specific requirements);
• A modelling procedure consists of the modelling activities to be performed in order to reach modelling goals; it may take the form of method documentation or may be supported by mechanisms aiming to improve user experience (e.g., by automating certain procedure steps).

![Diagram of modelling method building blocks](image)

Fig. 1.6 The modelling method building blocks [51]

Methodological agility is enabled by an iterative engineering process at the core of the AMME framework and depicted in Fig. 1.7. This process is generically named the “Produce-Use” cycle, with two phases per iteration: (1) the Produce step
will capture domain knowledge ("models of concepts"), formalise it and deploy it in a modelling tool; (2) the *Use* step will employ this modelling tool to capture case knowledge that instantiates the domain concepts ("models using concepts") while also evaluating acceptance and various quality criteria to feed back in the next iteration of the *Produce* phase.

**Fig. 1.7** The AMME Framework (adapted from [9])

This cycle may be conveniently specialised for different contexts and deployments. The assumption is that different instances will be necessary depending on the requirements to the conceptualization process. The "AMME Lifecycle" described below (Fig. 1.8) shows how a concrete instance of the conceptualization process is realized within the Open Models Laboratory (OMiLAB).

- **Create**: a mix of knowledge acquisition and requirements elicitation techniques;
- **Design**: practices for designing the modelling method building blocks depicted in Fig. 1.6;
- **Formalise**: refinements of the method design in terms of appropriate formalisms, to supporting implementations across various platforms by removing ambiguities from the method design specification;
- **Develop**: the modelling tool development phase, typically benefitting from rapid prototyping environments (e.g., [47]);
- **Deploy/Validate**: the packaging and deployment of the tool with improved user experience and an evaluation protocol that feeds back into the *Create* step of the next iteration.
Feedback loops occur both internally, between subsequent phases, and for the overall cycle, as each deployment collects change requests for the next method increments.

The Produce-Use cycle interacts, at the method “front-end”, with (1) the enterprise environment by assimilating requirements and domain knowledge; and, at the method “back-end”, with (2) an asset repository where lessons learned, method fragments and various reusable assets are accumulated for future deployments.

1.4 Project Experience and Results

1.4.1 The Open Models Initiative Laboratory

The Open Models Initiative Laboratory (OMiLAB) [10] is a research environment (both physical and virtual) that fosters a global community of researchers sharing a common understanding of the concept of modelling method and of models value. OMiLAB may be considered an instance deployment of AMME, providing specific enablers. A number of domain-specific or hybrid modelling methods and their deployments (tools) have been developed in projects of different kinds (1) educational (e.g., modelling tools for didactic purposes), (2) research-oriented (i.e., results of metamodelling tasks in research projects) and (3) digitisation-oriented (i.e., typically follow-up developments of research projects). A selection of such methods are presented in [52]—a first volume in a planned community-driven book series, reporting on projects that benefit from the OMiLAB enablers and its collaborative network.

Additionally, community-oriented events have established forums for dissemination or knowledge transfer between academic research, industry and education. The most prominent event is NEMO (Next-generation Enterprise Modelling)—an annual summer school [53] where the principles and framework of AMME
have been initially articulated and students have received initial training with its application. Currently OMiLAB has European and Asian “collaboratories”, as well as Associated Organisations fostering localised communities. An organisational structure and related management policies (e.g., for intellectual property rights) may be consulted in [54].

One key enabler provided by AMME is ADOxx—the rapid prototyping platform for developing and deploying modelling tools [47]. Its meta-metamodel provides built-in facilities for developing the building blocks of a modelling method—e.g., a design environment for the language grammar and vocabulary, a vector graphics language for dynamic notations, a scripting language for developing model-driven functionality. In addition, a richness of plug-ins and ancillary development services and reusable items are made available through the OMiLAB portal. Research is underway regarding MM-DSL, a platform-independent declarative language for modelling method definitions—an initial version was presented in [55].

### 1.4.2 The ComVantage Research Project

ComVantage, an FP7 European Project [56], proposed an IT architecture based on mobile app ensembles consuming Linked Enterprise Data shared among organisations, in support of collaborative business processes for various application areas (e.g., customised production, mobile maintenance) [57]. The run-time architecture was complemented with design-time support in the form of the evolving ComVantage modelling method, a process-centred enterprise modelling method tailored for the domain-specificity of the project application areas, for the goal of establishing a knowledge repository in diagrammatic form (hence supporting KM and EAM).

Various semantic lifting approaches were applied to unify heterogeneous data sources in a Linked Data cloud, from which front-end apps can retrieve them through protocols established by the Linked Data technological space [58]. An RDFizer mechanism was implemented to also expose the diagrammatic contents in Linked Data form [59, 60], thus contributing to the knowledge processes proposed in Fig. 1.3 and opening new opportunities of semantic lifting (as suggested in Fig. 1.9).

Consequently, requirements on client applications would inherently propagate to requirements for the modelling method, reclaiming an AMME approach to evolve it accordingly, and to ensure that a sufficient semantic space is available to clients.

By the end of the project, the modelling method reached a Zachman-style multi-viewpoint structure addressing various aspects (perspectives) and scopes (levels of domain-specificity) as reflected in Table 1.1. Multiple sources may be consulted for the method documentation [11, 61–63]. Details on the method’s conceptual evolution with respect to AMME are available in [41]. The modelling tool is hosted by OMiLAB at [12].
Table 1.1 Viewpoints of the ComVantage method [33]

<table>
<thead>
<tr>
<th>Scopes:</th>
<th>Aspects:</th>
<th>Structural aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Behavioural aspect</td>
<td>Structural aspect</td>
</tr>
<tr>
<td>Business/Enterprise</td>
<td>Procedural views: (procedural knowledge captured in the form of flowcharts with varying semantics and customised notation)</td>
<td>Collaborative views (the same kind of knowledge, expressed as interactions in order to highlight necessary interfaces)</td>
</tr>
<tr>
<td></td>
<td>Models that describe a business on its highest level of business process abstraction (e.g., abstract value creation processes)</td>
<td>Models that describe the values that are created by the business or by each process in particular</td>
</tr>
<tr>
<td>Requirements</td>
<td>Models that describe how work processes are mapped on requirements for different kinds of resources</td>
<td>Models that describe how different resources must interact based on their mappings on work processes</td>
</tr>
<tr>
<td>App execution</td>
<td>Models that describe how mobile apps must be “orchestrated” (chained) according to the flow of the process they must support</td>
<td>Models that describe mobile apps that are required and must be “orchestrated” to support a process</td>
</tr>
<tr>
<td>App design</td>
<td>Models that describe the flow of interactions between a user and elements of an app’s user interface</td>
<td>Models that describe the features and data requirements for a mobile app</td>
</tr>
</tbody>
</table>
1.4.3 The FCML/BEE-UP Educational Project

FCML (Fundamental Conceptual Modelling Languages) is a teaching-oriented modelling method providing a hybridisation of 5 well-known modelling languages: BPMN, EPC, ER, UML and Petri Nets. Their initials form the acronym BEE-UP which is the name of the modelling prototype made available through OMiLAB for teaching purposes, already adopted for teaching MDSE and BPM topics by several universities associated with the OMiLAB collaboration network. Details on the FCML method can be consulted in [13], only a brief overview is provided here.

FCML is not only a convenience tool that supports model types belonging to different languages. It also agilely assimilated semantic integration, extensions and functionality to address modelling requirements for various teaching scenarios subordinated to BPM (e.g., process path simulation, Petri Nets simulation) or MDSE (e.g., SQL code generation):

- In the “mechanisms” building block, all three types of mechanisms are exemplified: (1) generic (e.g., model queries or diagram exports in the form of RDF knowledge graphs), (2) specific (e.g., SQL generation from ER diagrams, Petri Nets simulation/stepping), (3) hybrid (applicable to different types of models complying to some well-formedness requirements—e.g., process path analysis...
for models that correctly use the basic workflow patterns, i.e., BPMN, EPC, UML activity diagrams—as suggested in Fig. 1.10;)

- In the “language” building block, semantic extensions are applied to support these mechanisms: (1) at language level (e.g., an organigram model type to support workload simulations); (2) at model type level, (e.g., EPC extensions to support multiple variants of EPC recommended in the literature, depending on their goal and required rigor); (3) at concept level (e.g., user-editable and machine-readable properties such as costs, times, probabilities to support process path simulations, SQL-specific properties to support SQL code generation);

- The “modelling procedure” component is aligned accordingly to guide users in how to create models for the different scenarios.

FCML and its BEE-UP implementation enable a multi-purpose and multi-layered modelling approach, providing on one hand notational alternatives for BPM and, on the other hand, a complementary set of languages for teaching MDSE topics. The modelling tool is hosted by OMiLAB at [14].
1.5 Summary and Future Challenges

The relevance of Conceptual Modelling to selected fields of research and management practices—namely, Business Process Management, Enterprise Architecture Management, Knowledge Management (Systems) and Model-driven Software Engineering—was hereby discussed. A common underlying requirement for modelling method agility was highlighted and the AMME framework was proposed as a complement to standard methods, which are typically considered invariants in agile development practices. Thus, the work at hand raises the level of agility from that of software engineering to that of modelling method engineering—even in the case of MDSE, where agility is advocated here in relation to generic “conceptual model”-awareness concerns (rather than standard-driven code generation). Experiences and results accumulated through the Open Models Initiative Laboratory research environment validate the applicability of AMME. The current experience is based on several project-based deployments and further enablers must be developed, similarly to how the agile software development practices have been emerging as a community-driven paradigm.

Several key enablers that must further consolidate the AMME vision are suggested here as open challenges for which research is already under way: (1) an executable declarative modelling language for coding modelling method definitions in a platform-independent manner; (2) interoperability mechanisms at meta²model level between the popular metamodelling platforms; (3) specialised issue tracking platforms considering the specific characteristics of modelling methods as Design Science artefacts.

References