Hans-Georg Fill
Harald Kühn *Editors*

Metamodeling: Applications and Trajectories to the Future

Essays in Honor of Dimitris Karagiannis

Springer

Metamodeling: Applications and Trajectories to the Future

Hans-Georg Fill • Harald Kühn
Editors

Metamodeling: Applications and Trajectories to the Future

Essays in Honor of Dimitris Karagiannis

Springer

Editors
Hans-Georg Fill
University of Fribourg
Fribourg, Switzerland

Harald Kühn
BOC Product & Services AG
Vienna, Austria

ISBN 978-3-031-56861-9 ISBN 978-3-031-56862-6 (eBook)
https://doi.org/10.1007/978-3-031-56862-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

This book is about the remarkable career of Prof. Dr. Dimitris Karagiannis, an academic researcher and teacher at the University of Vienna working in the general area of conceptual modelling. Dimitris is unique among academic colleagues because of his focus on high-impact concrete outcomes for his work that benefit the conceptual modelling community of researchers, practitioners, and users.

Throughout much of his career, Dimitris concentrated his efforts on four projects:

- Research on metamodelling and its application to conceptual model building. Metamodels are useful in conceptual modelling because they serve as blueprints of models in a given domain, from which one's own model can be derived through instantiation with less effort and better results than starting from scratch. For this project, Dimitris worked with his students and collaborators, resulting in the ADOxx meta-modelling platform for implementing domain-specific modelling tools (https://adoxx.org/) that is used by researchers and practitioners around the world.
- The founding and nurturing of the BOC Group (https://www.boc-group.com/en/), a technology company that is world leader in Enterprise Architecture (EA) tools with a special focus on BPM, EAM, and GRC as well as related services with thousands of customers and multiple awards from technology watchers. BOC was initially launched with tool prototypes developed in Dimitris' Department Knowledge Engineering (University of Vienna) and grew as businesses realized that the concepts and tools of EA constitute an effective way to design, manage, and evolve a business.
- The Open Models Initiative Laboratory, aka OMiLAB (https://www.omilab.org/), is an academic lab whose aim is to make accessible to the community models and modelling tools. In a similar sense, open source software is serving the software engineering community. OMiLAB is fulfilling its mission with tutorials, workshops, and a repository of modelling tools.
- The NEMO Summer School (https://nemo.omilab.org/) is a summer school on conceptual modelling, attended by doctoral students from around the world,

offering a comprehensive series of tutorial lectures by top researchers. NEMO will celebrate its 10th anniversary in 2024 and is generally regarded internationally as offering the best comprehensive introduction to conceptual modelling for young researchers.

The contents of the book include multiple chapters on metamodelling, the BOC Group technologies, and the OMiLAB, authored by Dimitris' students and collaborators. It also contains chapters by several presenters at the NEMO Summer School as a sample of the teaching material of the school. In reading the chapters, the reader gets an idea of Dimitris' focus on concrete outcomes in his research on metamodelling, BOC Group's approach to developing ground-breaking technology by exploiting metamodelling ideas, and the OMiLAB's approach to community building.

The impact of an academic's research is usually measured by citations and a researcher's h-index. But these are narrow metrics that measure impact solely through citations by other researchers (citation-impact) and ignore the impact of one's research on students, resulting from teaching activities (education-impact), for which Dimitris has the NEMO Summer School to show, but also countless other teaching accomplishments on conceptual modelling and metamodeling. In addition, impact of his work includes impact on the world through usage of one's ideas (usage-impact). If one were to include in the definition of impact all three metrics—i.e., citation-impact, education-impact, and usage-impact—Dimitris' research record would shine as one of the most influential among researchers in the history of conceptual modelling with only a handful of colleagues able to come close to his impressive and well-rounded accomplishments. This is why this book is worth reading.

Toronto, Canada John Mylopoulos
December 2023

Preface

o. Univ. Prof. Prof. h.c. Dr.-Ing. Dimitris Karagiannis has made fundamental contributions to research, education, and industry in the domains of Metamodeling, Conceptual Modeling, Enterprise Modeling, and Method Engineering. He is a very inspiring thought-leader and an exceptional personality. With this Festschrift, we would like to honor him for his guidance, inspiration, and dedication throughout many years. We are very grateful having the possibility to continually work with him since long time, both in academia and in industry. Dimitris Karagiannis supervised all authors in their early careers either during their PhD thesis or habilitation process.

Selected milestones of his remarkable career are as follows: Dimitris Karagiannis received his PhD degree in computer science in 1987 from the Technical University of Berlin. After heading the division "Enterprise Information Systems" at the Research Institute for Applied Knowledge Processing (FAW) at Ulm, Germany, from 1988 to 1992, he was appointed as full professor for Business Informatics at the University of Vienna in 1993, leading the Research Group Knowledge Engineering. In 1995, he founded the university spin-off BOC Group which has grown since into an international group in the areas of BPM, EAM, and GRC. BOC Group is the creator and vendor of the industry-leading products ADONIS, ADOIT, and ADOGRC as well as their underlying metamodeling platform ADOxx. Since 2005, Dimitris Karagiannis is the Chairman of the Supervisory Board of BOC Group. He founded the metamodeling community ADOxx.org as well as the Open Models Initiative Laboratory (OMiLAB.org), which is today an international network and community of renown research institutions and companies collaborating on the topic of modeling method engineering, enterprise modeling, and metamodeling. Since 2014, the OMiLAB network organizes an annual summer school on Next-Generation Enterprise Modeling (NEMO). In 2011, Dimitris Karagiannis was awarded the title of Professor honoris causa from the Babeş-Bolyai University, Cluj-Napoca, Romania for his outstanding contributions to Business Process Management and Business Informatics.

This book contains fourteen, single-authored essays from close collaborators of Dimitris Karagiannis from academia, research, and industry. Each chapter honors the extraordinary inspiration that Dimitris Karagiannis provided during the remarkable and ongoing collaboration. The essays are presented alphabetically by the last name of each author.

Franz Bayer applies metamodeling concepts in the domain of internal developer platforms and cloud platforms to foster business agility as well as technology agility.

Robert Buchmann elaborates on the concept of semantic-driven systems engineering, which has the goal of replacing traditional, mainly syntactic formats for model-driven engineering with ones that also convey semantics.

Hans-Georg Fill introduces the concept of spatial conceptual modeling, which allows knowledge expressed through conceptual models to be anchored in the real world using augmented reality technology.

Anke Helmes discusses from a practitioner's perspective the challenges of managing business processes and capabilities together and the added value of using them in a unified, integrated framework.

Knut Hinkelmann shows how ontology-based metamodeling can overcome the shortcomings of traditional model-driven engineering approaches in application development, knowledge-based systems, model validation, and knowledge management.

Florian Johannsen shows how metamodeling and conceptual modeling can be used to address the issues of smart services and environmental sustainability in quality management.

Harald Kühn illustrates observations made in the use of metamodeling platforms in industry and discusses expected developments in their architectures, in the continuous modeling method operation, and in their integration with LLM systems.

Christian Lichka introduces a metamodel-driven approach for a cross-domain, unified data and software architecture that adds a performance dimension to governance, risk, and compliance.

Christoph Moser addresses the challenge of democratizing enterprise architecture by making established modeling methods such as ArchiMate more accessible to a wider audience.

Martin Nemetz reflects on the concept of metamodeling and how it has been used to support intellectual capital reporting. He also presents two use cases for metamodeling in industry and identifies opportunities for further research.

Christoph Prackwieser explores the progression from traditional process simulation techniques to the advanced concept of digital twins within the context of business processes and supply chain management.

Wilfrid Utz presents a novel approach to method engineering that uses haptic design interactions for domain experts to derive formalized metamodel skeletons and digital prototypes for immediate testing and evaluation.

Robert Woitsch discusses how process-oriented knowledge management has evolved over time, reflecting on a series of research projects and suggesting support through an enabling IT infrastructure.

Srdjan Zivkovic provides an overview of modular metamodel engineering, introduces a comprehensive set of metamodel composition operators, and explores potential applications of microservices and AI to advance metamodeling practices.

Prof. John Mylopoulos is a shining light for all of us in the field of conceptual modeling. He has made groundbreaking contributions especially in the areas of artificial intelligence, information systems, and software engineering. To have him contribute the foreword to this book is an extraordinary honor, and we are deeply grateful to him.

Fribourg, Switzerland
Vienna, Austria
January 2024

Hans-Georg Fill
Harald Kühn

Contents

Contributors

Franz Bayer BOC Group, Vienna, Austria

Robert Andrei Buchmann Faculty of Economics and Business Administration, Babeș-Bolyai University, OMiLAB Node @ UBB-FSEGA, Cluj-Napoca, Romania

Hans-Georg Fill University of Fribourg, Fribourg, Switzerland

Anke Helmes Munich, Germany

Knut Hinkelmann FHNW University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland
OMiLAB gGMBH, Berlin, Germany

Florian Johannsen Faculty of Computer Science, University of Applied Sciences Schmalkalden, Schmalkalden, Germany

Harald Kühn BOC Products & Services AG, Vienna, Austria

Christian Lichka BOC Group, Vienna, Austria

Christoph Moser BOC Group, Vienna, Austria

Martin Nemetz Buchs, Switzerland

Christoph Prackwieser Adoplus Consulting Pty Ltd, Erskineville, NSW, Australia

Wilfrid Utz OMiLAB NPO, Berlin, Germany

Robert Woitsch BOC Products & Services AG, Vienna, Austria

Srdjan Zivkovic BOC Information Technologies Consulting, Vienna, Austria

How Metamodeling Concepts Improve Internal Developer Platforms and Cloud Platforms to Foster Business Agility

Franz Bayer

Abstract Business Agility is a crucial aspect of modern organizations, reflecting their ability to adapt and thrive in a rapidly changing business landscape. At its core, Business Agility involves being responsive to change, allowing companies to swiftly adjust their strategies and operations in response to market shifts, emerging technologies, and evolving customer demands. Developer Platforms play a pivotal role in fostering Business Agility. These platforms provide a foundation for building and deploying applications, enabling developers to collaborate seamlessly and iterate rapidly. Utilizing metamodeling concepts and agile methodologies throughout the stages of creation, design, formalization, development, deployment, and validation enables the introduction of a generic metamodel for Developer Platforms. This approach enhances the adaptability to changes and technological advancements. The essential elements of this metamodel are detailed in this paper, underscoring the significance of conceptual groundwork as a mission-critical step prior to initiating the orchestration or implementation of the Developer Platform's actual services and tools.

Keywords Metamodeling · Method engineering · DevOps · Internal developer platform · Cloud platform · Business agility · Developer experience

1 Business Agility for Digital Products

1.1 The Essence of Business Agility

Business Agility is a cornerstone of success on today's dynamic markets and especially on marketplaces for digital products and services. It enables organizations to adapt to changing market conditions, to consider new opportunities, and navigate

F. Bayer (✉)
BOC Group, Vienna, Austria
e-mail: franz.bayer@boc-group.com

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_1

unexpected challenges with resilience. Focusing on innovation, user-centricity, and a culture of continuous improvement, Business Agility requires operational efficiency to accelerate time-to-market for products and services [1].

Business Agility enables organizations to quickly respond and adapt to changes in the market, technology, or regulatory landscape. This flexibility is essential for staying competitive and thriving in unpredictable conditions. Being agile allows businesses to better meet customer needs and expectations. By staying responsive to customer feedback and market trends, organizations can tailor their products and services to deliver greater value, leading to increased customer satisfaction and loyalty [2]. With short and incremental development and feedback cycles, product developers can identify and address potential issues early in the process, reducing the likelihood of major drawbacks [3].

Within the organization, Business Agility needs a culture of innovation and creativity [4]. Agile workplaces empower developers by giving them more autonomy, fostering a sense of ownership, and encouraging collaboration. This leads to higher levels of job satisfaction and engagement, ultimately contributing to increased productivity. Through regular reflection, adaptation, and learning from experiences, developers can refine their processes, enhance performance, and stay ahead in an ever-evolving market for digital products.

1.2 Holistic View on Digital Products as a Service

Digital products are nowadays offered mainly as a service, delivered through a subscription-based model or an on-demand access. Ultimately, a digital product is no longer a set of features and some basic non-functional requirements but transformed into a complex service with a substantial focus on availability, maintainability without downtimes, responsibilities, liabilities, security, compliance, confidentiality, audit-readiness, support communication, reporting and support time. On top of this additional scope, user-centricity led to an additional focus on user- and customer-touchpoints to have immediate feedback on the delivered services at any time throughout the entire service life cycle [5].

The development organization's response to this shift in paradigm was the establishment of tool chains, policies, and heavy weight processes. Complicated build chains, costly release processes and eventually the mismatch with deployment and operations requirements led to DevOps principles and the motivation to consider the holistic requirements of a Software-as-a-Service (SaaS) product development and its operation [6]. This was a big milestone in managing the complexity of state-of-the-art software development. But along with all advantages, some challenges and risks were created as well. The most important challenges are related to the resistance to change, the complexity of the tool chain, the maintenance efforts in general and the cognitive load for developers. Unfortunately, these challenges cannot be addressed on a long run by organizational changes or improving the productivity of the development teams with more advanced agile methods based on scrum, extreme programming, or the like. The root cause of these challenges are typically a combination of a rigid product and service strategy, monolithic architectures, cumbersome release cycles and a low maturity in DevOps excellence [7].

1.3 Technology Agility

Technology Agility is a pivotal aspect of modern business operations and as a part of Business Agility, marked by the seamless integration of architecture vision, continuous delivery, and DevOps excellence [8] to overcome a waterfall-based build chain approach. Architecture vision serves as the guiding force, offering a comprehensive blueprint that aligns technological initiatives with organizational goals. This foresight ensures that systems are not only robust but also flexible, adapting to evolving requirements and technological advancements (Fig. 1). Continuous delivery, a key aspect of Technology Agility, facilitates the swift and reliable release of software updates, enabling organizations to respond promptly to changing market demands [9, 10]. DevOps excellence further enhances agility by fostering collaboration and automation across development and operations teams, streamlining workflows, and accelerating the delivery pipeline. In concert, these elements empower businesses to navigate the dynamic landscape of technology with agility, fostering innovation and responsiveness in an ever-evolving digital ecosystem [11].

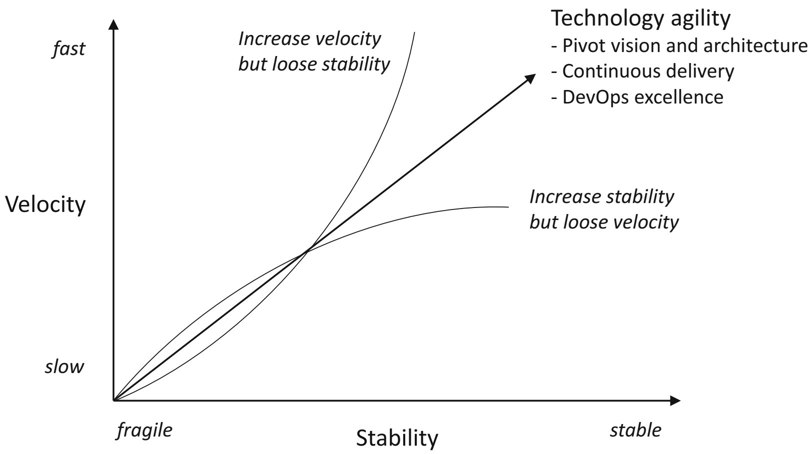


Fig. 1 Elasticity of velocity and stability to manage Technology Agility according to [8]

2 Internal Developer Platforms and Cloud Platforms

Developer Platforms play a pivotal role in shaping the future of technology, serving as the foundation upon which innovation thrives. With robust and user-friendly Developer Platforms, developers are equipped to break barriers, transcend limits, and build the next generation of groundbreaking applications.

In essence, motivating Developer Platforms is about fueling the passion and ingenuity that progress, ensuring that developers have the tools they need to turn their visions into reality [12]. Two different categories of Developer Platforms need to be considered when analyzing the potential as well as the current limitations. A Cloud Platform is commonly associated with a Platform-as-a-Service (PaaS), where the runtime is overseen by the vendor whereas an Internal Developer Platform (IDP) is typically grounded on-premise or in an Infrastructure-as-a-Service (IaaS) environment and overseen by the organization itself (Fig. 2).

2.1 Internal Developer Platforms

An IDP is a tailored set of tools, practices, and infrastructure provided by an organization to its internal development teams. It aims to streamline software development by offering a centralized environment with standardized tools, frameworks, and libraries. This promotes consistency, compliance, and best practices across the organization [11]. Standardized development tools, frameworks, and libraries play a crucial role by ensuring consistency and minimizing time spent on individual environment configurations. Automation is a key component of an IDP, incorporating automated build and deployment processes, CI/CD pipelines, and testing to streamline the entire life cycle and reduce manual tasks. An IDP, whether on-premise, cloud-based, or hybrid, provide IaaS, allowing development

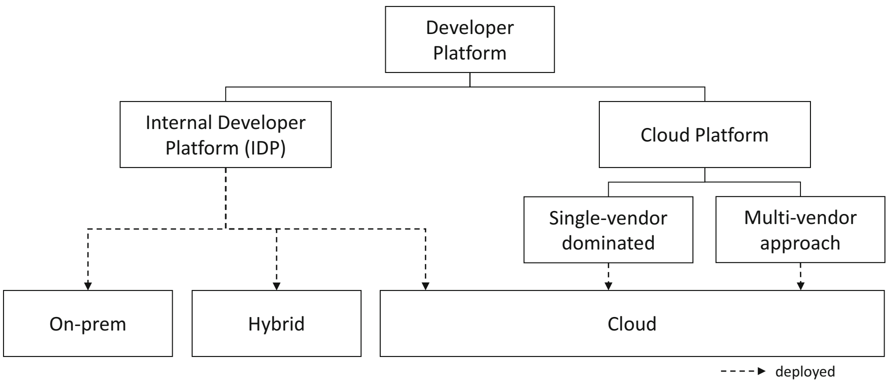


Fig. 2 A categorization approach for Developer Platforms

teams to concentrate on application creation without worrying about infrastructure provisioning. Collaboration is facilitated through integration with version control systems, communication platforms, and collaboration tools, promoting seamless teamwork. Security features, code analysis, and quality assurance are embedded in the platform, supported by documentation and examples to ensure adherence to best practices. An IDP often expands to include tools for monitoring application performance, issue tracking, and log collection, enabling rapid issue identification and resolution. Ultimately, by offering a centralized and standardized development environment, an IDP contributes to accelerated development cycles, enhanced collaboration, and efficient resource utilization within the organization.

2.2 Cloud Platforms

A cloud platform incorporates a unified set of services and infrastructure components for building, deploying, and managing applications in the cloud. Provided by cloud service providers as PaaS, these platforms offer a spectrum of services, from basic computing to advanced features like serverless computing, hybrid databases, data analytics, IoT, and machine learning. They enable on-demand access to computing resources over the internet, facilitating dynamic scaling without the complexity of infrastructure management. Serverless computing allows users to run code without server provisioning, ensuring automatic scaling based on demand, with users paying only for actual compute resources used. Cloud platforms also include services for secure user identity and access management, along with security measures like encryption, firewalls, and monitoring tools to protect against unauthorized access and cyber threats. Additionally, they provide tools for monitoring application performance, health, and analytics services for extracting insights from data. Popular cloud platforms include Amazon Web Services (AWS) [13], Microsoft Azure [14], Google Cloud Platform (GCP) [15], IBM Cloud [16], and Oracle Cloud [17]. Organizations leverage these platforms to build, deploy, and scale applications more efficiently, benefiting from the flexibility, scalability, and cost-effectiveness of cloud computing.

2.3 Business Agility Requirements Mapped to Developer Platforms

Business Agility and Developer Platforms are closely linked, with cloud computing playing a crucial role in boosting the agility of modern businesses [18]. Developer Platforms facilitate swift provisioning of computing resources, essential for scaling based on varying workloads or evolving business needs. This agility allows businesses to respond promptly to changes in demand, ensuring efficient

Table 1 Business agility requirements mapped to IDPs and cloud platforms

Business agility	IDP	Cloud platform
Strategic alignment with pivot capability and high demand for change responsiveness.	Change responsiveness is depending on the integration capabilities of the IDP. High efforts for maintainability.	Cloud platforms provide a flexible and dynamic environment, allowing businesses to adapt quickly to changes. Costs need to be managed actively.
Iterative and incremental value proposition, evaluation, continuous reassessment, and feedback loops.	High maturity and knowledge of continuous delivery, monitoring, and data-analytics capabilities are required. Hyper-automation is a key aspect.	Basic built-in continuous delivery and monitoring capabilities are available. Usually, data-analytics based on the business model needs to be considered individually.
Autonomous teams with cross functional collaboration and end-to-end view on the service life cycle as well as on the customer touchpoints over time.	Collaboration capabilities and CI/CD help to keep track on the entire life cycle. Documentation, examples, and trainings need to be provided individually. Development principles need to be governed individually.	Learning and improvement is typically part of the PaaS offering. Cloud platforms offer collaboration services and governance, but also artificial intelligence or serverless computing. Cross functional teams can experiment with new technologies without significant upfront investments.
Predictability, antifragility, resilience, and knowledge of scalability requirements.	Scalability and resilience need to be incorporated in the design, implemented, and controlled tightly, but cause high efforts.	Reference architectures and implementations provide an out of the box scalability but with an impact on the cost allocation for the SaaS model.

resource allocation and service delivery without delays. Table 1 summarizes the basic business needs and how they map to either an IDP or a Cloud Platform.

A mature Developer Platform is a fundamental enabler of Business Agility by providing the technological foundation for change responsiveness, innovation, rapid deployment, and cost efficiency. Even if Cloud Platforms are nowadays able to provide a rich foundation, the holistic view on a Developer Platform is a complex system, and obviously, the design, implementation and operation need a systematic approach to meet the outlined requirements.

3 Method Engineering for Technology Agility

Conceptual modeling methods are fundamental in managing complexity by employing abstraction for specific purposes. A comprehensive modeling approach encompasses a modeling language, a modeling procedure, and associated mechanisms and

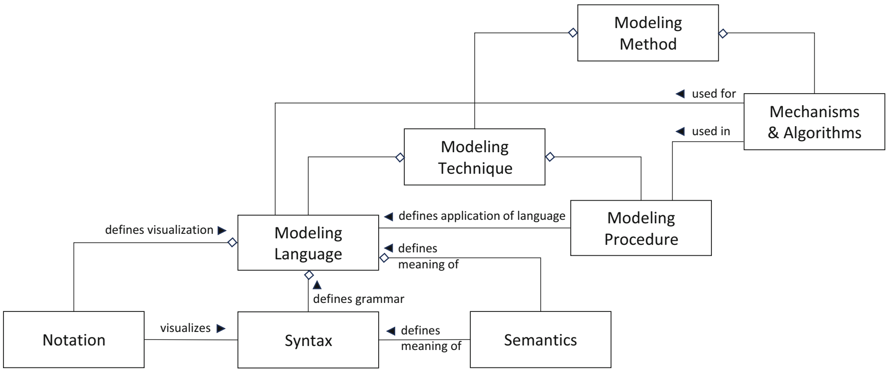


Fig. 3 Components of a modeling method [19]

algorithms (refer to Fig. 3 [19]). The core foundation of a modeling method is the modeling language, which can be further divided into its syntactic elements (syntax), graphical representation (notation), and meaning (semantics). The modeling procedure outlines the steps taken by modelers and the outcomes achieved when applying a modeling language to generate valid models. Mechanisms and algorithms define the functionality, such as information retrieval, data acquisition, data analysis, simulation, and transformation, that method engineers need to implement in a corresponding modeling tool.

3.1 Agile Modeling Method Engineering (AMME)

Method Engineering for the design and implementation of a Developer Platform to foster Technology Agility utilize modeling methods not only for the orchestration or implementation of model-driven systems or automation but initially for creating abstract representations with the purposes of understanding and communication, as highlighted by Mylopoulos [20]. An additional emergent requirement is agility, allowing metamodel concepts, building blocks and tool implementations to evolve and adapt to changing method engineering needs.

The Agile Modeling Method Engineering (AMME) framework, as outlined in [21, 22], advocates for an iterative life cycle depicted in Fig. 4. This framework encompasses an engineering cycle that commences with knowledge acquisition and requirements analysis (the create phase) and culminates in the deployment of a functional modeling tool (the deploy phase). The intervening stages involve design (yielding a specification of the modeling method building blocks), formalize (resulting in a formalism-oriented specification), and develop (resulting in an implemented modeling tool). Method engineers can incorporate micro iterations between the design and develop phases for swift adjustments, bypassing other phases.

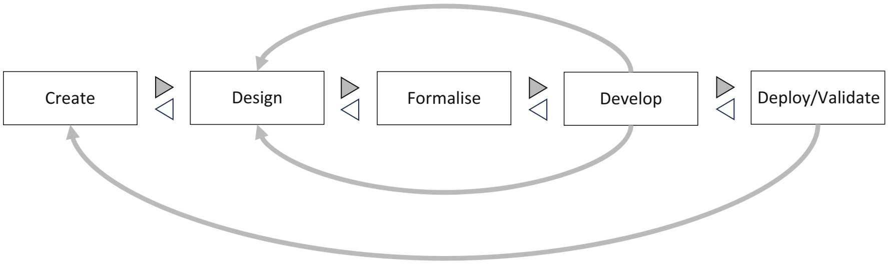


Fig. 4 AMME Framework and agile life cycle [22]

The AMME framework's iterative and incremental nature fosters agility by anticipating the evolution of modeling requirements, like the way agile software development principles arose from the need for responsiveness and manageable granularity in software engineering [23]. As depicted in Fig. 4, a feedback loop exists between this engineering cycle and an evaluation cycle, grounded in hands-on experience with the deployed prototype.

3.2 Metamodel for the Design of Developer Platforms

In the past, software development excellence relied on the creativity and conceptual strength of teams, operating within transparent abstraction layers and implicit supply chains. These skills remain pivotal in today's software development, although in a completely different environment.

The evolving abstraction layers and supply chain act as both enablers and sources of complexity. Many companies attempt to navigate this complexity through tool implementation and the adoption of best practices. However, for some, this strategy results in substantial overspending on resources and money for tool chain and technology management. Ultimately, excessive focus on internal housekeeping contradicts to the ability to swiftly adapt to changing business requirements and align software products with market needs. To solve this problem, a simple Metamodel (Fig. 5), developed with the utilization of the AMME Framework, can be used as a blueprint to work on the concepts before investing in the development and maintenance of a Developer Platform. Every element within the metamodel represents a crucial facet of a Developer Platform and should be contemplated at a conceptual level. The subsequent sections outline these concepts and offer insights that must be considered for the successful implementation of the platform.

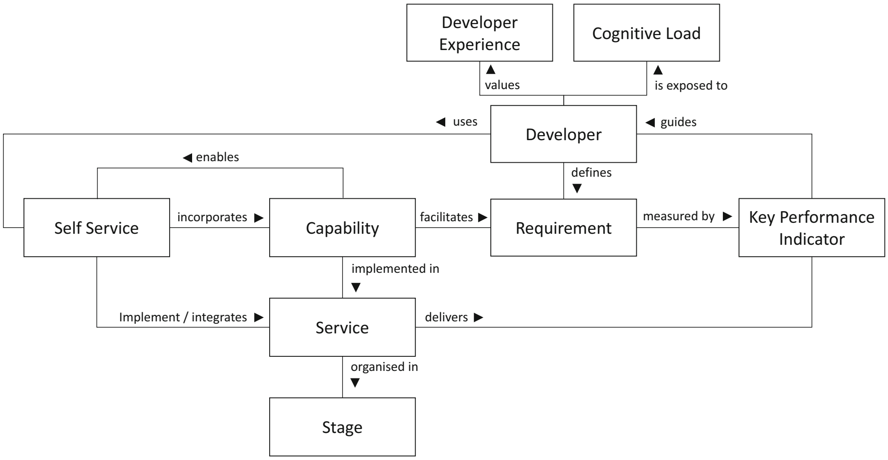


Fig. 5 A metamodel for design, build, operation, and maintenance of a Developer Platform

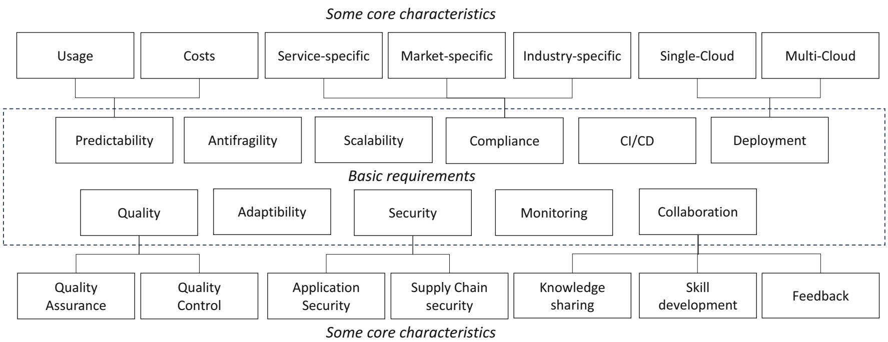


Fig. 6 Basic Requirements for a Developer Platform with some core characteristics

4 Requirements

Commencing with the establishment of prerequisites for designing and implementing a Developer Platform, the software architecture and technology stack play a crucial role in determining differences, whether in scope or required maturity level for each identified requirement. Nevertheless, in a second step, a generalization can be applied to compile a map of requirements which can be used as a baseline for further identification of requirements or a deep dive into a single requirement.

Figure 6 outlines such a map of requirements. The basic requirements can be directly assigned to the metamodel presented in Fig. 5, whereas the presented core characteristics are just examples of details for some of the basic requirements to demonstrate the further drill-down which might be necessary when it comes to a specific software architecture or technology stack. Already in this early phase of method engineering based on the Metamodel outlined in Fig. 5, not only the scope of the requirements is important but also the expected maturity level and related Key Performance Indicators (KPIs). It is highly recommended to define some KPIs in an early stage of the process to make sure that an agile, incremental design and implementation can be monitored. With a clear picture of the requirements, the method engineer can choose, enrich, and identify capabilities for the Developer Platform.

5 Capabilities and Self-Service

Typically, DevOps and Agile Best Practices deliver a set of standard capabilities to be utilized for the implementation on a Developer Platform. A set of examples is outlined in Fig. 7.

The provided capabilities are on the one hand side a big advantage but to some extent also a risk to fail because of three basic issues that need to be solved:

- *Alignment of requirements with capabilities* to ensure that the capability not only addresses or enhances the requirement at a local level but also contributes to a comprehensive end-to-end life cycle.

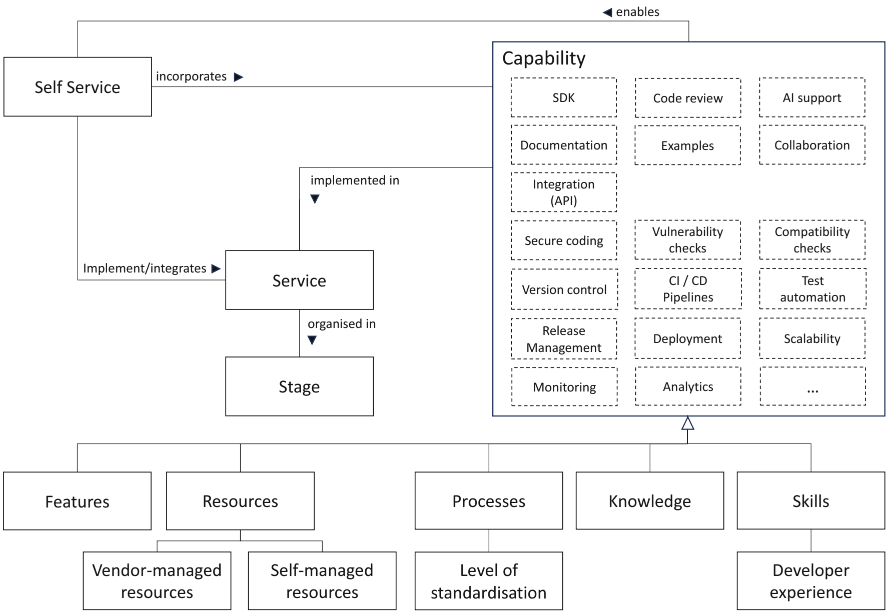


Fig. 7 Extending the Capability of the Metamodel presented in Fig. 5

- *Composition and Assessment of Capabilities* and not only the integration of tools to facilitate informed decisions and change responsiveness.
- Foster DevOps principles with the support of *Self-Service as a Capability* to compose, create, extend, or change capabilities.

The following examples aim to inspire and raise awareness about the three issues, encouraging the utilization of metamodeling concepts for designing Developer Platforms. This approach eventually fulfills the needs of Technology Agility and, in turn, nurturing Business Agility.

5.1 Alignment of Requirements with Capabilities

A Developer Platform encompasses a range of capabilities designed to facilitate and enhance the software development process. Some key capabilities of a Developer Platform are outlined in Fig. 7. Every capability required to construct a Developer Platform is interconnected and represents not only a set of features but a systematic and coordinated management of features, resources, processes, knowledge, and skills. In the process of aligning requirements with capabilities, all these aspects of a capability need to be aligned. Otherwise, a perfect new feature will be integrated in the capability map, but the organization has no knowledge or skills to integrate the processes into the existing ones. Or maybe even the processes coming with the capability are not fitting the overall requirements for Quality, Compliance or Security.

5.2 Composition and Assessment of Capabilities

The composition of capabilities is a key aspect to achieving change responsiveness and informed decision-making as soon as new capabilities are onboarded or changes with a high impact need to be considered.

Figure 8 shows an example of such a composition. The capability "*Secure Coding*" needs to be integrated in an existing Capability map where "*Quality*" with its detailed capabilities already exists. When analyzing the capabilities, it is discovered that the capability "*Code analysis*" exists in both. To further analyze the Capabilities, the Metamodel in Fig. 7 can be used to determine the features, resources, processes, knowledge, and skills already existing because of the existence of "*Quality*" and how to integrate the new features, resources, processes, knowledge, and skills needed for "*Secure Coding*". The composition of capabilities using the proposed conceptual integration allows the evaluation of effectiveness and maturity. This assessment helps identify strengths, weaknesses, and areas for improvement, allowing the organization to prioritize its efforts in the creation, re-composition, or change of capabilities.

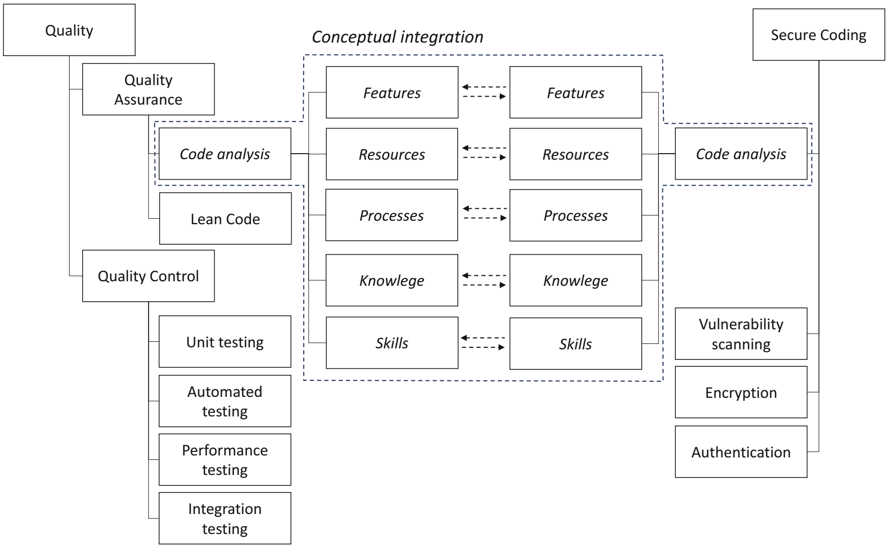


Fig. 8 Conceptual integration of two Capabilities "*Quality*" & "*Secure Coding*"

5.3 Self-Service as a Capability

To cater to the requirements of Technology Agility, a Developer Platform should incorporate self-service capabilities. This encompasses the provision of a portal service equipped with tools, resources, and documentation, enabling developers to independently explore, comprehend, and leverage the capabilities and underlying services of the platform. Such a portal offers an easy way to support the use case of capability composition presented in Fig. 8 on a conceptual level before the orchestration or implementation of services is triggered. Working on the level of capabilities enables the developer to instantiate the capability for every new integration with other capabilities and manifest this conceptual integration in a template. Next to the single capabilities also the templates will be provided in the self-service portal for further re-use. Once a template is released, the integration on the underlying services can be done. Eventually, a specific capability as well as a native capability for further integrations will be available in different templates (Fig. 9).

6 Service Orchestration and Stages

Developer Platform services orchestration involves the coordinated management and integration of various tools and processes within a development environment to streamline and automate the software development life cycle. This orchestration encompasses workflow automation, service integration, API management,

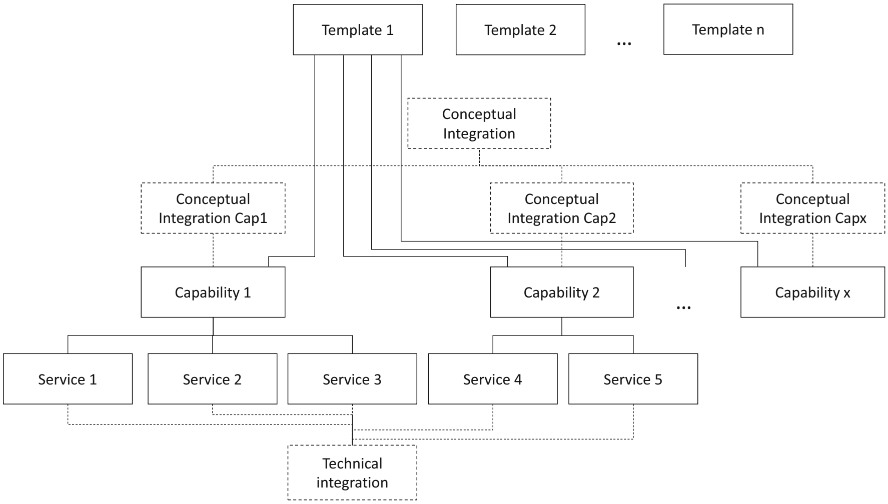


Fig. 9 Templates: conceptual level (capabilities) and technical integration (services)

containerization, microservices orchestration, event-driven architecture, security measures, collaborative development tools, feedback loops, and environment provisioning [24].

Service orchestration of the Developer Platform involves managing the state of the entire process. The orchestrator maintains information about the current state of the workflow and uses this information to make decisions about the next steps in the process and different stages. In addition, the stages outlined in Fig. 10 require long-running Processes [25]. Orchestrating services allows for greater scalability and flexibility in the overall system. New services can be added, and existing ones can be modified or replaced without disrupting the entire workflow. Monitoring tools can track the progress of each service, providing insights into performance, bottlenecks, and overall efficiency.

7 Developer Experience and Cognitive Load

Developer experience (DX) and cognitive load are interconnected concepts in software development, both playing crucial roles in influencing the productivity and satisfaction of developers. DX is the overall experience that developers have while using tools, frameworks, libraries, and other resources in the process of software development. A positive DX contributes to increased productivity, faster development cycles, and a more enjoyable work environment for developers.

Cognitive Load, on the other hand, refers to the mental effort required for a person to complete a particular task. It is divided into three main types:

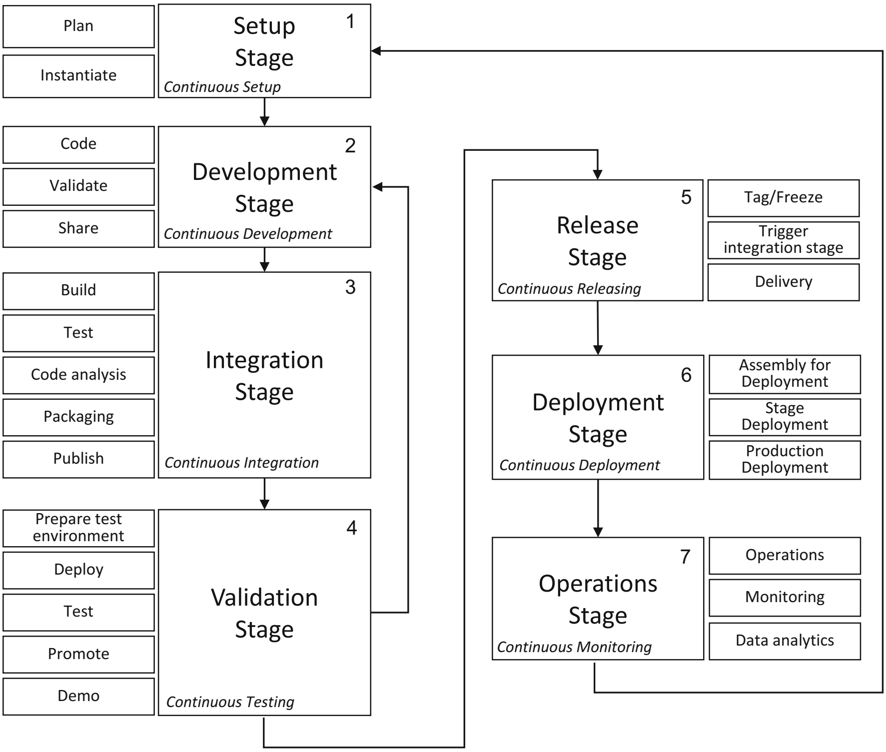


Fig. 10 Service orchestration along the stages of a developer platform

- *Intrinsic Cognitive Load* refers to the inherent difficulty of a task.
- *Extrinsic Cognitive Load* refers to the extra cognitive burden placed on individuals due to the way a task is presented or organized.
- *Germane Cognitive Load* pertains to the cognitive load associated with the organization and comprehension of information, playing a role in learning and problem-solving.

Good DX aims to reduce the cognitive load associated with using a Developer Platform. Intuitive interfaces, clear documentation, and consistent design contribute to lower extrinsic cognitive load. Developers can focus more on the intrinsic cognitive load of solving complex problems rather than struggling with the tools themselves [26] (Table 2).

A positive DX minimizes the learning curve for new tools and technologies, reducing the extrinsic cognitive load during onboarding. Well-structured documentation, tutorials, and examples contribute to a smoother learning experience, lessening the cognitive load associated with acquiring new skills. Timely and informative feedback from development tools helps reduce the cognitive load associated with debugging and troubleshooting. Clear error messages, effective

Table 2 Developer experience [27]

Aspect	Meaning	Characteristics
Conation	How do developers see the value of their contribution?	Goals, plans, alignment Intention, motivation, commitment
Affect	How do developers feel about their work?	Respect, team, social Attachment, belonging
Cognition	How do developers perceive the development infrastructure?	Platform, techniques, process Skill, procedures

logging, and interactive debugging tools contribute to a more efficient resolution of issues, minimizing frustration and cognitive load. Consistency in design, behavior, and conventions across different tools and components contributes to a lower extrinsic cognitive load. Predictable behaviors help developers form mental models, reducing the cognitive load associated with understanding how tools and systems work.

8 Developer Platform Metrics and Key Performance Indicators

DevOps metrics and Key Performance Indicators (KPIs) are essential for measuring the effectiveness, efficiency, and overall performance of a Developer Platform [28]. These metrics help teams understand their progress, identify areas for improvement, and make data-driven decisions to optimize the software development and delivery process. Table 3 shows the essence of these KPIs. Whereas *Deployment Frequency* and *Lead Time for Changes* indicate velocity, *Mean Time to Recover* and *Change Failure Rate* indicate stability, both requirements introduced in Fig. 1.

Deployment Frequency represents the number of deployments or releases within a specific timeframe (e.g., weekly, or monthly). Increased deployment frequency indicates the ability to deliver changes quickly and respond to user needs.

Lead Time for Changes represents the time it takes for a code change to go from commit to production. Short lead times signify efficient development and delivery processes.

Mean Time to Recover (MTTR) represents the average time it takes to restore service after a production incident or outage. A lower MTTR indicates effective incident response and resolution capabilities.

Change Failure Rate represents the percentage of changes or deployments that result in a failure. A low change failure rate reflects the reliability and stability of the deployment process.

Additional KPIs must be considered to elevate the maturity level of the Developer Platform and cultivate increased responsibility among all developers.

Deployment Success Rate represents the percentage of successful deployments compared to the total number of deployments. A high deployment success rate

Table 3 Software delivery performance: DORA metrics [29]

KPI	How to interpret the KPI?	Target of elite organizations
Development frequency	For the primary application or service, you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deployments per day)
Lead time for change	For the primary application or service, you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production?	Less than one hour
Time to restore service	For the primary application or service, you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour
Change failure rate	For the primary application or service, you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0–15%

indicates the reliability of the deployment pipeline and the effectiveness of the Developer Platform capabilities.

Code Churn represents the frequency and extent of code changes or modifications. Monitoring code churn helps identify areas of high development activity, which may require additional attention for quality assurance and quality control.

Test Automation Coverage represents the percentage of automated tests covering the codebase. Higher test automation coverage ensures more comprehensive testing and faster feedback for quality control.

Incident Volume and Severity represents the number and severity of incidents reported. A decrease in incident volume and severity indicates improved system stability.

Mean Time Between Failures (MTBF) represents the average time between system failures. A higher MTBF suggests increased system reliability and resilience.

Work in Progress (WIP) represents the number of tasks or user stories in progress at any given time. Monitoring WIP helps manage workloads and ensures a balanced flow of tasks.

Deployment Cost represents the cost associated with each deployment. Reducing deployment costs supports efficiency and resource optimization.

Customer Satisfaction tries to get some indicators using surveys, feedback, or Net Promoter Score (NPS) related to the delivered service. Positive customer feedback reflects the impact of DevOps practices on end-user experience.

Employee Satisfaction tries to get some indication related to job satisfaction using Employee surveys or feedback. Satisfied and engaged teams are more likely to contribute to a successful DevOps culture.

It's essential to tailor the selection of metrics and KPIs to align with the specific goals, priorities, and context of the organization. Regularly reviewing and adjusting these metrics allows teams to adapt and continuously improve their Developer Platforms and DevOps practices.

9 Conclusion

The design, build, operation, and maintenance of a Developer Platform goes beyond the classic tool implementation projects. It requires a strong focus on the conceptual work upfront to avoid monolithic, rigid, and complex tool chains which are difficult to manage and by no means responsive for changes. The metamodel outlined in Fig. 5, along with the more detailed perspectives in the subsequent sections, aims to provide insight and the foundation for further investment in conceptual aspects. This could involve the introduction of roadmap planning, investment planning, change expectation planning, maturity assessments, and even risk management for the Developer Platform. Once implemented, every facet of the platform contributes to the formation of a responsive service ecosystem built on continuous life cycle management. The ability to adapt to changes is crucial for business success and significantly impacts the overall prosperity of a software development organization.

References

1. Pichler, R.: Strategize - Product Strategy and Product Roadmap Practices for the Digital Age. Pichler Consulting (2016)
2. Kim, G., Behr, K., Spafford, G.: The Phoenix Project - A Novel About IT, DevOPs, and Helping Your Business Win. IT Revolution Press (2013)
3. Waster, T.E.: Value Proposition Design - How to Create Products and Services Customers Want. Wiley (2014)
4. Gray, D., Vander Wal, T.: The Connected Company. O'Reilly Media (2012)
5. McQuivey, J.: Digital Disruption: Unleashing the Next Wave of Innovation. Amazon Publishing (2013)
6. Smart, J., Berend, Z., Ogilvie, M., Rohrer, S.: Sooner Safer Happier - Antipatterns and Patterns for Business Agility. IT Revolution Press, Portland, OR (2020)
7. Newman, S.: Monolith to Microservices - Evolutionary Patterns to Transform Your Monolith. O'Reilly Media (2019)
8. Enterprise Business Agility Model: https://agilityhealthradar.com/enterprise-business-agility-model/. Accessed 12 Jan 2024
9. Humble, J., Farley, D.: Continuous Delivery - Reliable Software Releases Through Build, Test, and Deployment Automation. Pearson Education Inc. - The Addison Wesley Signature Series (2011)

10. Wolff, E.: Continuous Delivery - Der pragmatische Einstieg (in German). dpunkt.verlag (2015)
11. Kim, G., Humble, J., Debois, P., Willis, J.: The DevOps Handbook - How to Create World-Class Agility, Reliability & Security in Technology Organizations. IT Revolution Press (2021)
12. Operating Model for Agile: Organizations. https://businessagility.institute/. Accessed 12 Jan 2024
13. AWS Cloud Products: https://aws.amazon.com/products/. Accessed 12 Jan 2024
14. Azure Cloud Resources: https://azure.microsoft.com/en-us/resources/developers. Accessed 14 Jan 2024
15. Google Cloud Platform: https://console.cloud.google.com/. Accessed 14 Jan 2024
16. IBM Cloud: https://www.ibm.com/cloud/. Accessed 20 Jan 2024
17. Oracle Cloud: https://www.oracle.com/cloud/. Accessed 21 Jan 2024
18. Hohpe, G.: Cloud Strategy - A Decision-Based Approach to Successful Cloud Migration. Gregor Hohpe AWS Wroclaw Poland (2021)
19. Karagiannis, D., Kühn, H.: Metamodeling platforms. In: Bauknecht, K., Min, T.A., Quirchmayr, G. (eds.) Proceedings of the 3rd International Conference EC-Web (LNCS 2455). Springer, Berlin (2002)
20. Mylopoulos, J.: Conceptual modelling and telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modelling, Databases, and Case: An Integrated View of Information System Development, pp. 49–68. Wiley, New York (1992)
21. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics, pp. 5–10 (2015)
22. Karagiannis, D.: Conceptual modeling methods - the AMME agile engineering approach. In: Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W. (eds.) Domain Specific Conceptual Modeling - Concepts, Methods and ADOxx Tools, pp. 3–21. Springer (2022)
23. Karagiannis, D., Buchmann, R.A., Burzynski, P., Reimer, U., Walch, M.: Fundamental conceptual modeling languages in OMiLAB. In: Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.) Domain Specific Conceptual Modeling - Concepts, Methods and Tools, pp. 3–30. Springer (2016)
24. Cambell, E.: Microservices Architecture: Make the Architecture of a Software as Simple as Possible. CreateSpace Independent Publishing Platform (2012)
25. Newman, S.: Building Microservices - Designing Fine-Grained Systems. O'Reilly Media (2021)
26. Fagerholm, F., Münch, J.: Developer experience: concept and definition. In: Proceedings of the International Conference on Software and System Process (ICSSP 2012), Zurich, Switzerland, June 2–3, pp. 73–77 (2012)
27. Lawson, J.: Ask Your Developer: How to Harness the Power of Software Developers and Win in the 21st Century. Harper Business (2021)
28. Radford, P., Cleary, D., Lawrie, R.: Software Metrics and Measurement Case Studies. Charismatek Software Metrics (2012)
29. DORA Metrics: https://dora.dev/research/. Accessed 12 Jan 2024

Semantics-Driven Systems Engineering: Requirements and Prerequisites for a New Flavor of Model-Driven Engineering

Robert Andrei Buchmann

Abstract Model-driven engineering (MDE) traditionally relied on a generic notion of "model transformation" to bridge design with software implementation, or the so-called "design-time" and "run-time" phases of systems engineering. Frameworks to support this have been standardized, leading to formal specifications such as MOF or XMI. The field is dominated by XML serializations for perhaps historical reasons, at a time when XML seemed to be an interoperability paradigm more than a (meta)language and was holistically adopted for all model interchange and bridging concerns in MDE.

This chapter discusses an emergent notion of "Semantics-driven systems engineering" (SDSE) as a specific flavor of MDE and argues that knowledge representation must take over the role traditionally allocated to XML, as custom semantics must be made available to engineering processes and digital artifacts by means of knowledge capture tools. SDSE may be superseded by the generic MDE notions if we abstract away its pragmatic characteristics; however, this chapter advocates that it is worth examining its specialized characteristics having both methodological and technological specificity. They can inform research and practice on requirements and pre-requisites to potentially streamline SDSE as a repeatable process and as a practice expanding the value proposition of domain-specific conceptual modeling, to be highlighted here through examples derived from OMiLAB educational or research projects.

Keywords Model-driven engineering · Knowledge graphs · Agile modeling method engineering · Domain-specific modeling · RDF · OMiLAB

R. A. Buchmann (✉)
Faculty of Economics and Business Administration, Babeș-Bolyai University, OMiLAB Node @ UBB-FSEGA, Cluj-Napoca, Romania
e-mail: robert.buchmann@econ.ubbcluj.ro

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_2

1 Introduction

Model-driven engineering (MDE) [1], in its traditional interpretation, relies on a generic notion of "model transformation" to turn diagrammatic conceptual models into running code, or to turn models into other models—further subjected to processing for design-time purposes (e.g. simulation) or to bindings to a technological environment (i.e. platform-specific models). In terms of standards, this has been supported by formal specifications such as MOF [2] or XMI [3], with XML schemas being the dominant way to go for standard model serializations, storage or interchange. The adoption of XML as an interoperability paradigm has led to competing schemas, e.g. XPDL, BPMN XML for business process representation, or schemas that evolved to address diverging requirements, e.g. BPEL to bridge business processes with Web services.

However, conceptual models, including domain-specific or platform-specific ones, are fundamentally used to capture knowledge and therefore should benefit from means of knowledge representation feeding into knowledge-driven engineering processes or artifacts. Repurposing data exchange formats to achieve machine-readability of models seems somewhat forced when knowledge representation formats, just as stable and robustly formalized, have been available for a while, typically converging to graph structures (semantic networks, conceptual graphs) that are closer to the "boxes-and-arrows" typically involved in diagrammatic modeling or knowledge visualization. Refactoring diagrammatic structures to fit the XML DOM tree may not be very difficult but adds an extra layer and transformation effort—which may be hidden to some extent by tools, but this also takes away freedom of modelers or engineers to flexibly query and incorporate domain-specific semantic networks into the model-driven engineering processes. The term (model) "query" is often superseded by the dominant but generic notion of "model transformation". Notable exceptions can be found in the field of business process management where there's been long term interest in process querying methods [4], and a few experimental query languages such as GMQL [5] or VMQL [6], relying on subgraph searching and matching.

At the convergence of interoperability standards (including here the semantic interoperability aspect) and subgraph searching/matching there lies at least one robust standard—the Resource Description Framework (RDF) [7], complemented by the industry-driven CypherQL for labelled property graphs [8], and potentially leading to a unifying Graph Query Language standard [9].

Based on this observation, we've introduced in [10] the notion of Linked Open Models, where the Linked Open Data principles for distributed graph data were repurposed for (a) diagrammatic content, (b) model annotations (including data structures attached to model elements e.g. for simulation purposes or as runtime traces), and (c) model links commonly used for inter-model navigation. Essentially this offers the possibility to treat diagrammatic models for any domain as RDF graphs, including the possibility to link model elements to existing ontologies or external live resources, to incorporate them into rule-based systems and to expose them to SPARQL clients.

At that time, the work emerged in a limited project scope—i.e. as part of the Comvantage domain-specific modeling method, based on model transformation patterns reported in [11] and initially implemented for the ADOxx metamodeling platform [12]. Later this was incorporated in the Bee-Up educational tool [13], thus offering the possibility to apply the Linked Open Models treatment to BPMN models (a direction also pursued by other researchers in [14]), to UML models (a direction also pursued via labelled property graphs in [15]), to Petri Nets, DMN models, EPC models and others. As the design space offered by the Bee-Up tool iteratively expands, its semantic enrichment features have been presented as tutorials in several conferences. e.g. ER 2023 [16]. Outside the Bee-Up tool, a generalized value proposition was enabled by offering the ADOxx-to-RDF convertor as an ADOxx plug-in, thus facilitating its adoption across many projects—the plug-in can be obtained at [17] and was used to demonstrate OWL reasoning patterns over a domain-specific language, in [18].

From the recurrent application of the plug-in across different application areas, a repeatable engineering process took shape, which is generalized here under the label of *Semantics-driven Systems Engineering* (SDSE). Requirements, pre-requisites, and further opportunities for consolidating this as a specific flavor of MDE will be gleaned in the following sections, based on observations on how this engineering process played out in two types of contexts: (a) in student projects, as the ADOxx-to-RDF convertor was adopted to address the "conceptual modeling education design problem" formulated in [19] from a Design Science perspective and motivated a novel teaching method [20]; students are regularly using this in knowledge engineering tasks, exploiting the interplay between domain-specific modeling and knowledge graphs, some of their work being reported in a number of student research papers [21]; (b) in research projects, demonstrating cross-domain applicability and advocating a long term shift of MDE tools towards a graph-based treatment of model contents [18]. Further reports on employing knowledge representation in model-driven engineering processes are expected to derive from the dedicated workshop series KG4SDSE—Knowledge Graphs for Semantics-driven Systems Engineering [50].

The remainder of this paper is structured as follows: Section 2 will highlight the characteristics of SDSE based on a minimalist deployment example. Section 3 will summarize past cases and projects that contributed to synthesizing SDSE. Section 4 will derive requirements and pre-requisite tooling that must be organized to deploy SDSE as a streamlined method. The paper ends with a concluding discussion in support of a shifting role for conceptual modeling towards future information systems.

2 Characteristics of Semantics-Driven Systems Engineering

We are experimenting with SDSE in the context of the OMiLAB Digital Innovation Environment [22]—a teaching and demonstrator installation incorporating several toolkits, robotic devices and adapters for model-driven engineering or operation.

One of the components of this installation is the "cooking robotic arm" that can interact with modeling environments or knowledge structures in various ways, as suggested by Fig. 1, which points to three flavors of model-driven engineering. All three rely on a flowcharting DSML (domain-specific modeling language) that extends classic flowcharting to describe both platform-specific atomic abstractions—robotic movements—and, aggregating on top of those, scenario-specific abstractions—cooking recipes and resources (ingredients, tooling). The physical ingredients are images on laminated cards to be picked by the robotic arm and stacked on a plate according to the order of cooking steps in a recipe. The three demonstrated flavors of MDE are:

(a) **Traditional MDE**, where specific constructs of the DSML are converted into parameterized executable code, i.e. into HTTP requests to trigger the robotic

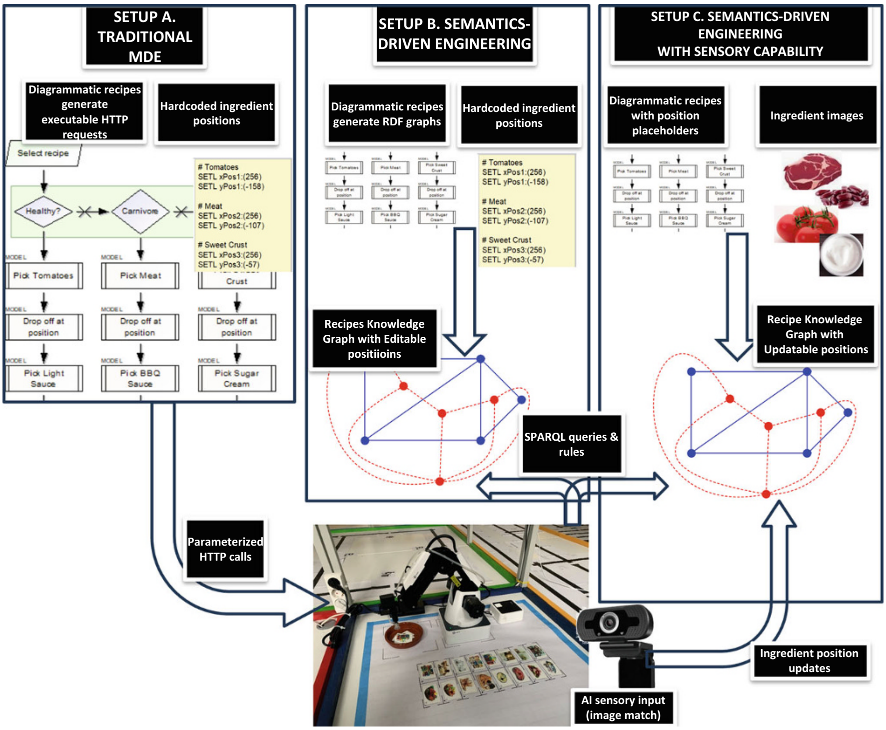


Fig. 1 From traditional MDE to SDSE

movements. Ingredient positions are hardcoded as diagram annotations. The user gets to select the preferred recipe based on decision nodes present in the flowchart.

(b) We label as **semantics-driven engineering** the configuration where the cooking recipes are converted into a knowledge graph (RDF-based) governed by the DSML metamodel and additional annotations. It stores ingredient positions as data properties in ingredient descriptions linked to the recipe steps, besides quantities, types, and any other property—captured either as diagram annotations or added afterwards into the graph itself. This knowledge graph is exposed to SPARQL queries to identify at run-time the recipe steps, ingredient requirements or incompatibilities, before the actual robotic movement requests are executed on the ingredient positions. The user provides domain-specific constraints and rules and based on those recipe matching is performed—Fig. 2 suggests how a DSML can describe user preferences, to be matched by SPARQL against the knowledge graph to get matching recipe suggestions. User-provided rules may enforce semantic validation (e.g. are the vegan recipes really avoiding animal by-products?), retrieve relationship-driven information

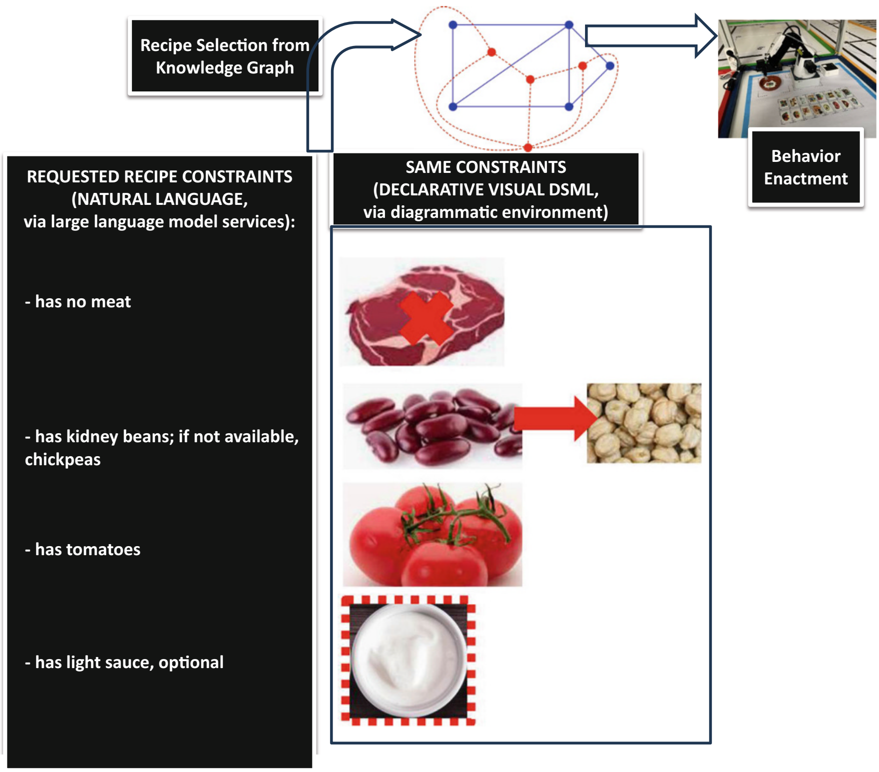


Fig. 2 Domain-specific semantic input from the user (natural language or diagrammatic)

(which are the recipes using ingredient X after ingredient Y?), compute relevant aggregations (build the list of ingredients and the total cost for a selected meal). Large language model services can be involved to collect user preferences or queries via natural language—a feature popular with recent versions of semantic graph databases [23], although a loss of reliability manifests due to the stochastic nature of such services and further experimentation is needed to assess the effectiveness of prompt engineering approaches, depending on the choice of ontology, domain-specific jargon etc.

(c) Semantics-driven engineering can be further complemented by **sensory capabilities** if a smart Webcam service performs image recognition and updates the ingredient positions accordingly, rather than having them manually edited or hardcoded in diagram annotations. This leads to a hybrid symbolic-subsymbolic service orchestration where ad-hoc sensory data complements the persistent rule-driven knowledge base. The configuration mimics the slow thinking/fast thinking duality [24], between decisions made on a priori knowledge structure (i.e. before the cooking starts) and run-time decisions involving sensors (e.g. ingredient missing during cooking).

Based on this minimalist configuration, one can infer what differentiates the generic notion of "model transformation" according to traditional MDE from semantics-driven engineering—i.e. the MDE reliance on a fixed, often standard, metamodel subjected to stable mappings to platform-specific executable abstractions. In SDSE, an evolving DSML, potentially also expanded by ad-hoc linking to external taxonomies, gradually incorporates the necessary domain-specific types and relationships by means of metamodeling, to ensure sufficient competency for the expected semantic queries and reasoning rules. Model contents are employed as a knowledge base to be queried by run-time or configuration-time code, rather than as a more abstract/visual version of executable code. This is in line with earlier interpretations of being "model-driven"—i.e. models being used as blueprints by a human agent (software developer) who looks at them to understand design decisions they need to implement or enact; only this time, the "look and understand" effort is transferred to artificial agents with the help of semantic queries and purposeful domain-specificity that must be captured agilely, in a DSML responsive to competency requirements.

This reclaims a user-oriented (i.e. low code) metamodeling environment to support the semantic definition of types and relationships—including multi-hop relationships that get explicit meaning via rules; inferencing/constraining can also be added by advanced knowledge engineers via mechanisms based on OWL/SHACL, however modeling environments and CASE tools typically lack such features.

This is not to say that the traditional notion of "model transformation" from MDE does not already cover graph rewriting, matching, path finding, and other operations involved in semantic graph querying—but it is worth separating SDSE as a flavor of MDE that places more emphasis on management of semantics instead of management of code complexity, mappings and structure. Such a repeatable process is generalized in Fig. 3, which distinguishes the need for explicit stages

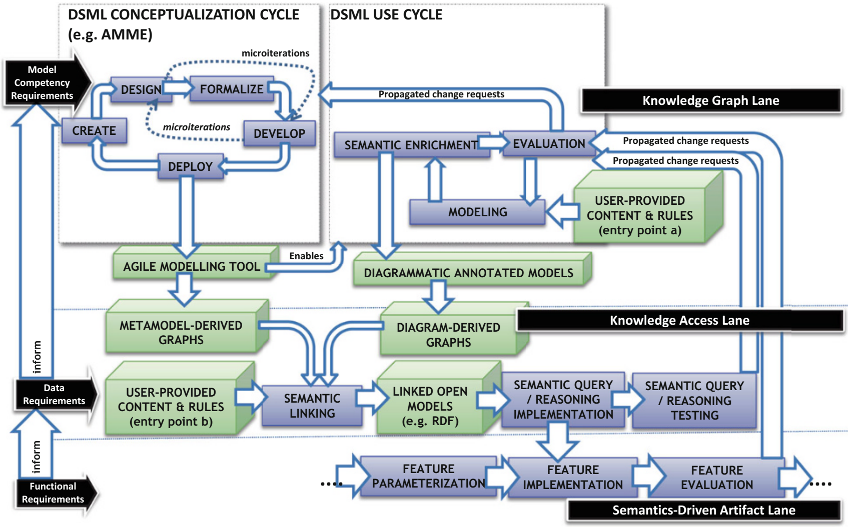


Fig. 3 High level view on SDSE as a repeatable process

dedicated to (a) DSML conceptualization by metamodeling (OMiLAB's Agile Modeling Method Engineering framework [25] is depicted for that part), (b) DSML use for diagrammatic knowledge capture and annotation, (c) knowledge graph derivation from the DSML metamodel, models, annotations and additional user-provided input, (d) semantics-driven feature parameterization, and (e) the feedback loops necessary to respond to evolving competency requirements—informed by the functional requirements of the final artifact.

3 Exemplary Cases

Several exemplary cases are referenced here and depicted in Fig. 4, where SDSE emerged as a recurrent pattern—particularly from a project management (i.e. task structuring) perspective, but also from a technological specificity perspective. These are student projects that started from developing DSMLs for narrow scopes and evolved towards building artifacts operated according to the semantics captured by those DSML and some auxiliary descriptions (e.g. diagrammatic annotations, natural language).

(a) In the project summarized by paper [26], the conceptualization of agile software projects informed a DSML for decomposing and describing a software project in terms of epics, user stories and a backlog items taxonomy. This was, on one hand, used to populate a Jira project with the diagrammatically defined roles and backlog; on the other hand, it was further linked to an existing BPMN and

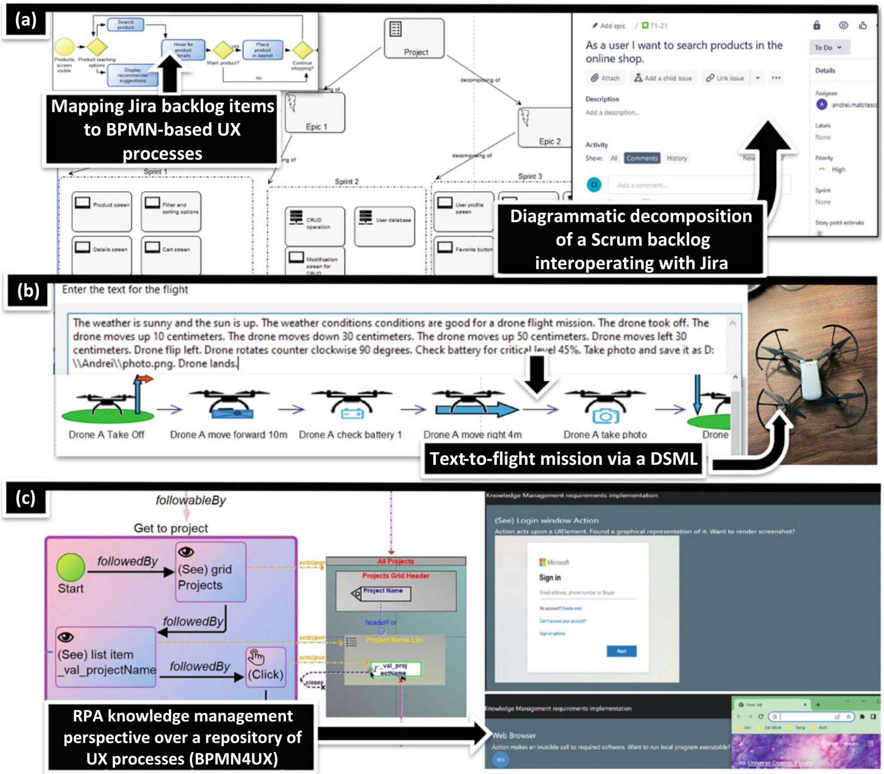


Fig. 4 Impressions of DSMLs and artifacts realized through SDSE, based on projects described in [26] (**a**), [27] (**b**) and [28] (**c**)

UML implementation to allow the mapping of backlog items to either structural components of the software project (UML components and use cases) or to the user interface processes where user stories belong (as a specialization of BPMN);

(b) In the project summarized by paper [27], a DSML was engineered to prescribe flows of drone actions, not unlike the cooking robot recipes introduced in the previous section. A Spacy-based interface "compiled" natural language descriptions of drone flight missions into the structured processes conforming the DSML, further enriched by technology-specific or mission-specific properties;

(c) In the project summarized by paper [28], BPMN was customized into a DSML for describing user experience processes consisting of granular user actions conforming a UX taxonomy and mapped on user interface elements. This supports a knowledge management approach to Robotic Process Automation projects where user interface processes must be re-engineered to facilitate automation.

4 Prerequisites and Requirements for a Streamlined SDSE

In order to support the characteristics described in Sect. 2, adequate methodological and technological/tooling support is needed, currently available as ingredients that can be repurposed from different fields—from DSML engineering platforms to graph database management systems, natural language processing toolkits or large language model services. This paper's proposal of an explicit flavor of MDE emerged by employing these ingredients for project-specific needs in articulated and repeatable engineering processes such as the one synthesized in Fig. 3.

A critical prerequisite is to have model storage and interchange delegated to knowledge representation and reasoning techniques, instead of the dominant XML-based formats imposed since the times of syntactic interoperability challenges. Even the current drafts of OMG's specifications, e.g. XMI, now rely on UUIDs as identifiers of model elements to expand the scope of their identification. This can also be achieved with the URIs employed by Linked Open Data, with the additional benefit of having all identities Web-scoped and benefitting from structured identifiers—e.g. for hierarchical namespacing, or the possibility of dereferencing model elements demonstrated in [29].

The second prerequisite is to hide as much as possible of the underlying technological specificity needed for model serialization and storage, so that citizen knowledge capture becomes possible with as little technological friction as possible. "Boxes-and-arrows" sketching techniques are quite common (e.g. mind mapping), storyboarding for design thinking is also increasingly popular [30]—the user experience for domain-specific semantics capture should not be much more complicated, benefitting from semantic transparency on graphical level to facilitate human memorability and communication.

A third key prerequisite is that the knowledge capture should also support conceptualization evolution, e.g. by combining the use of a DSML with metamodeling means that can adjust the modeling competency and semantic richness of that DSML. As Fig. 3 shows in the conceptualization cycle box, we employed in our experience with SDSE the Agile Modeling Method Engineering (AMME) framework [25], specifically devised to methodologically guide the evolution and fast prototyping of a DSML in response to requirements for an expanding semantic space and model competency.

Out of these prerequisites, one can generalize requirements for future fully-fledged platforms that may be dedicated to streamlining this as a development process, or even as a software product line. Therefore, the requirements listed below extrapolate beyond the concrete tooling on which the reported examples were based:

Requirement 1. Visual semantic network manipulation. Knowing subjects (i.e. domain experts, but also citizens that possess limited but relevant problem-specific knowledge), should be empowered to build knowledge graphs by means that are visually expressive and familiar to laypersons. Those means should not be too different from common mind mapping and "boxes-and-arrows" sketching, possibly even involving natural language input to the knowledge structuring environment.

Even in the absence of technical expertise related to ontology engineering or graph serializations, users must be able to create graph data structures and populate them with datasets if needed. The tradition of metamodeling can provide best practices and methodology for building visual tools that abstract away, in a "low code" manner, the technical machine-oriented details while ensuring dynamic, interactive, and semantically transparent notation in the sense of Moody's principles [31]. Multi-mode switching and a frictionless experience should allow users to avoid visual cluttering, hide details, establish links across multi-perspective diagrams, switch between boxes-and-arrows sketching and tabular editing. Currently most visual knowledge graph tools are either visualizers-only [32] or ontology editors tightly coupled to the OWL jargon [33]. The envisioned user experience should achieve what spreadsheets or visual forms achieved for tabular/relational database management, supporting citizens with both designing and populating knowledge structures, on top of powerful "under-the-hood" means for storing and exposing those structures to engineering processes.

Requirement 2. Under-the-hood semantic graphs. Model storage and interchange formats should move away from legacy data interchange solutions and from the associated retrieval mechanisms (XML with XPath, DOM parsing) towards technologies that are by default amenable to navigating semantic networks—i.e. semantic graph databases for which robust tooling is now widely available, with RDF and labelled property graphs as dominant solutions and a future GQL standard promising technological unification. This can bring the visual networked knowledge manipulation closer to the machine-level manipulation, while removing the artificial hierarchical abstraction layer imposed by the DOM parsers. It can also enable semantic linking and enrichment outside the visual environment, as in the repurposing of Linked Open Data principles for Linked Open Models [10] or for linking model elements to run-time traces [34]. This can further have implications on other related paradigms—e.g. contextualizing event logs by moving away from their XML representation, as recent research stresses the need for a semantic approach to process mining [35]. An intermediate abstraction layer may still be necessary to achieve a technology-agnostic user experience, but the maintenance of mappings and model transformations between graphs of different flavors (visual, RDF, LPG) should become more straightforward.

Requirement 3. Technology-agnostic user experience with platform-specific transformations. Abstracting away technical details also means avoidance, in the diagramming experience, of technology-specific jargon pertaining to the technical solution "under the hood". Avoiding RDF/OWL or Neo4J jargon is critical, allowing users to focus on building networks of associations with visual simplifications for complex patterns (blank nodes, n-ary relationships, RDF-star). While a complete avoidance of technical jargon may not be realistic, a Pareto principle of handling most graph structures with minimal technical jargon should be applied, gradually requiring technical expertise for corner, cases or as the citizen knowledge capture stage feeds into to the semantics-driven artifact engineering. A DSML can unify graph abstractions that are technology-agnostic and intuitive enough for a visual diagramming experience, while allowing for a choice of diverse transformations to diverse technologies—RDF serializations, CypherQL scripts etc. Semantic annotation and linking to resources outside the visual environment should be similarly supported.

Requirement 4. Granular knowledge edits. A seamless user experience also implies that granular edits must be synchronized between the visual environment and the semantic graph repository. Projects hereby referenced used an ADOxx plug-in for manual export/import of entire diagrams, even when a single element is edited. This was slightly improved in Bee-Up's implementation where the RDF generation works as simple as a *Save as* operation, but still produces a file to be manually loaded into a semantic graph database. A streamlined experience should maintain in synch the diagrammatic environment and the semantic graph storage, via the available interoperability interfaces offered by knowledge graph platforms (e.g. SPARQL HTTP protocol, Neo4J Bolt protocol). Actually, a graph database should be the primary model repository for any kind of diagrammatic model, rather than a secondary "export" feature or plug-in. Modeling tools should include features for semantic annotation, URI dereferencing or ontology alignment—to make it obvious that conceptual modeling is really about refining, communicating and managing semantics.

Requirement 5. Explicit management of modeling requirements. The knowledge graph that knowing subjects will build must be informed by explicit *competency requirements*—derived from the model-driven artifacts to be engineered and satisfied by the DSML used for knowledge capture. This has two critical implications:

- The first refers to sacrificing a traditional characteristic of ontologies, of representing a consensus (shared) conceptualization of a domain—which implies knowledge stability, reuse, but also rigidity in the face of evolving requirements. Instead, enterprise-specific or even project-specific small scale knowledge schemas get prioritized in SDSE, possibly hybridizing available ontology terms, but also improvising new constructs—perhaps oversimplified, agreed upon in narrow scopes, or selected based on purpose, not unlike preferring Schema.org concepts to achieve SEO benefits in the Web. Enterprise knowledge graphs may include enterprise-specific concepts derived from internal practices that are irrelevant outside that company or must be obfuscated against external agents. A notion of "personal knowledge graphs" [36] is also emerging, potentially involving subjective perspectives tightly coupled to personal purpose. This leads to a potentially explosive heterogeneity of conceptualizations—it can be mitigated by ontology alignment techniques, or it may be tolerated if democratization and decentralization is what we actually aim for in citizen knowledge capture spaces;
- The second implication is that dedicated means for managing the knowledge graph requirements must be devised, as a distinct class from the requirements pertaining to end user-facing software components. This class of requirements, occasionally referred to in this paper as "competency requirements", are dependent on functional or data requirements and should be traceable too those, but is typically neglected and very limited tooling and methodological support is

available for managing it. A proposal was made in the form of the CoChaCo modeling tool [37] which maps metamodel elements to stakeholder purposes and model-driven features built for those purposes. It was originally introduced for mapping DSML requirements in OMiLAB projects but it is equally applicable to mapping ontology constructs to competency questions and functional requirements pertaining to user-facing features. A fully-fledged production line for SDSE may even generate initial drafts of visual knowledge graphs or DSML metamodels out of such visually modelled requirements—CoChaCo has some documentation generation capabilities in this sense.

Requirement 6. Agile metamodeling. A framework for characterizing modeling languages, tools or methods was proposed under the name "Purpose-Specificity Framework" [38], where purpose and specificity are seen as orthogonal dimensions instead of the traditional dichotomy between "general purpose" and "domain-specific". DSMLs and their tools must be "agile" in the sense that they must shift in this space of purpose diversity (execution, simulation, configuration etc.) and specificity levels (technology-specific, case-specific, formalism-specific etc.) in response to changing requirements. Semantic queries and reasoning rules could have varying degrees of prescription—awareness of some schema parts, discovery of previously unknown relationships (e.g. as in DESCRIBE SPARQL queries), or a mix of "expected" and "discovered" properties to be navigated or filtered by the model-driven clients. The DSML employed for knowledge capture must be able to accommodate this need for flexibility, instead of imposing a rigid metamodel to prescribe some fixed mappings.

Requirement 7. Seamless metamodeling experience. In addition to metamodeling agility, it should be possible for arbitrary domain semantics to be added on-the-fly, by flexibly crossing over the traditional separation between MOF levels. This is achieved in recent versions of some metamodeling platforms (e.g. ADOxx starting with 1.8 allows metamodel editing in the modeling environment), in emerging platforms that are ontology-driven [39], or in paradigms that by default consider any number of abstraction layers, e.g. multi-level modeling [40]. RDF is also a framework that uses uniform graph-based representation for all layers and aspects involved in a knowledge graph—data structures, facts, ontology/schema, reasoning rules, validation rules—therefore it is suitable as a mediator between the user-centric environment for knowledge capture and the knowledge graph platform. Large language models may also contribute, since it is common for natural language discourse to switch between layers of abstraction even in the same phrase. The visual knowledge manipulation experience should offer the same freedom that RDF or natural language allows, while also maintaining the unambiguous separation wherever necessary.

Requirement 8. A societal meta-requirement on education: One critical societal requirement is to recognize conceptual modeling as a scientific discipline in its own right. The work of [41] showed that it manifests the key characteristics of an academic discipline, while in [19] we criticized the dominant curricular designs that scatters this discipline as chapters/tooling subordinated to other disciplines (software engineering, business process modeling, databases etc.) failing to reveal its intrinsic nature and benefits, as means of knowledge capture and structuring for any application area.

5 Concluding Discussion: Towards the Mediating Role of Conceptual Modeling

The recent literature on Information Systems research points to theory building potential regarding three aspects currently manifesting in what authors label as "digital worlds" or "digital-first" services/products, which may be the model-driven artifacts operated or engineered through SDSE:

I. The **ontological reversal** phenomenon, initially a figure of thought from E. Husserl complaining about natural scientists who take abstract models as more real than the systems they were supposed to be modeling. This was transferred by Information Systems scholars [42] to the modern digitalization-driven landscape, pointing to examples of "things" that emerge primarily in digital form and end up shaping physical reality and behavior, rather than digital models being representations of known systems. Digital Twins are also a manifestation of this phenomenon, where the traditional goal of complexity reduction gets traded for the goal of retaining sufficient and evolving complexity to achieve granular binding to properties found relevant in a Physical Twin.
II. The shifting role of conceptual modeling—**from representation to mediation**—identified in [43], can be seen as an immediate consequence of the ontological reversal. Certain kinds of conceptual models, e.g. digital twins, are not limited to serve the traditional purpose of human understanding and communication or to represent already manifesting behavior; instead, they are designed to manifest behavior into existence or, more precisely, to inform with sufficiently rich knowledge structures the enactment of behavior, both human and artificial—this can be seen as a large scale, knowledge-driven and high complexity version of how an "actuator" is used in Internet of Things devices.
III. Finally, **citizen-centric design spaces and processes** such as those discussed in [44] are necessary to democratize the new flavor of model-driven engineering. Democratization is already successful in process automation or IoT products, as laypersons are empowered to automate by intuitively guided visual means. Google Home products support customers with configuring a "home knowledge graph"; Siri Shortcuts and Google Assistant routines help orchestrate tasks on personal mobile devices; even Robotic Process Automation platforms support individuals with automating personal daily tasks. The "Low-code" paradigm [45] pushes this trend, while from the Semantic Web community we see emerging solutions for citizen-oriented management of social linked data [46] or personal knowledge graphs [36]. The two perspectives, process-focused and knowledge-focused, of "low code" are expected to converge into digitalization

platforms that will enable citizens to both express the semantics relevant to a problem's domain and to engineer the solution to that problem in terms of the captured semantics, possibly even transferring problem solving recipes between domains by switching the semantic conception of the problem domain.

SDSE may be a methodological thread fusing together these phenomena into new value creation practices for the digital-first world. The ontological reversal counts on the pre-requisite of having agile design means to capture the ontological structures and data models needed to bring artifacts and behavior into existence. The democratization of those means requires citizen-oriented knowledge engineering spaces to actuate behavior in digital artifacts. Conceptual modeling would be a core activity in those spaces, supporting *complexity governance* rather than *complexity reduction* and ultimately fulfilling the technical mediation role between human interpretation of a problem context and the machine understanding needed to solve that problem. Conceptual modeling education must play a key role in empowering citizens with the skillset needed to perform such work, and this requires treating conceptual modeling as a standalone discipline with dedicated teaching installations, a direction where the OMiLAB Digital Innovation Environment contributes [22].

It remains to be seen how Large Language Models will also contribute—both modeling by prompting [47] and model-driven prompt engineering [48] are being investigated, pointing to diverse possibilities of interplay between natural language interfaces and "boxes-and-arrows" knowledge capture means.

From a knowledge management perspective, this can also lead to updates to Nonaka's seminal knowledge conversion spiral, SECI (Socialization-Externalization-Combination-Internalization) [49], which must be revisited in hybrid intelligence settings that consider new knowledge conversion modes—not only human-to-human, but also human-to-artificial, artificial-to-human and artificial-to-artificial.

References

1. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice. Springer (2017). https://doi.org/10.1007/978-3-031-02549-5
2. Object Management Group: The meta-object facility specification. https://www.omg.org/mof/ (2023)
3. Object Management Group: XML metadata interchange. https://www.omg.org/spec/XMI/ (2015)
4. Polyvyanyy, A. (ed.): Process Querying Methods. Springer, New York (2022). https://doi.org/10.1007/978-3-030-92875-9
5. Delfmann, P., Steinhorst, M., Dietrich, H.-A., Becker, J.: The generic model query language GMQL—conceptual specification, implementation, and runtime evaluation. Inf. Syst. **47**, 129–177 (2015). https://doi.org/10.1016/j.is.2014.06.003
6. Störrle, H.: VMQL: a visual language for ad-hoc model querying. J. Vis. Lang. Comput. **22**(1), 3–29 (2011). https://doi.org/10.1016/j.jvlc.2010.11.004
7. W3C: The resource description framework. https://www.w3.org/RDF/ (2014)
8. Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Schuster, M., Selmer, P., Taylor, A.: Formal semantics of the language Cypher. https://arxiv.org/abs/1802.09984v2 (2018)

9. The GQL standard committee: GQL standard. https://www.gqlstandards.org/ (2023)
10. Karagiannis, D., Buchmann, R.A.: Linked open models: extending linked open data with conceptual model information. Inf. Syst. **56**, 174–197 (2016). https://doi.org/10.1016/j.is.2015.10.001
11. Buchmann, R.A., Karagiannis, D.: Pattern-based transformation of diagrammatic conceptual models for semantic enrichment in the web of data. Proceedings of KES 2015. Procedia Comput. Sci. **60**, 150–159 (2015). https://doi.org/10.1016/j.procs.2015.08.114
12. BOC GmbH: The ADOxx metamodeling platform. https://www.adoxx.org (2023)
13. OMiLAB NPO: Bee-up for education. https://bee-up.omilab.org/activities/bee-up/ (2023)
14. Bachhofner, S., Kiesling, E., Revoredo, K., Waibel, P., Polleres, A.: Automated process knowledge graph construction from BPMN models. In: Proceedings of DEXA 2022, LNCS 13426, pp. 32–47. Springer, New York (2022). https://doi.org/10.1007/978-3-031-12423-5_3
15. Leon, A., Santos, M.Y., Garcia, A., Casamayor, J.C., Pastor, O.: Model-to-model transformation: from UML class diagrams to labeled property graphs. Bus. Inf. Syst. Eng. (2023). https://doi.org/10.1007/s12599-023-00824-9
16. Buchmann, R.A., Burzynski, P., Utz, W.: Semantic enrichment & digital twins based on conceptual modeling: the Bee-Up tool. ER 2023 Tutorials Program. https://er2023.inesc-id.pt/program/tutorials/#tutorial3 (2023)
17. OMiLAB NPO: ADOxx RDF transformation service. https://code.omilab.org/resources/adoxx-modules/rdf-transformation (2023)
18. Karagiannis, D., Buchmann, R.A.: A proposal for deploying hybrid knowledge bases: the ADOxx-to-GraphDB interoperability case. In: Proceedings HICSS 2018. AIS eLibrary. https://aisel.aisnet.org/hicss-51/ks/ks_creation/4/ (2018)
19. Buchmann, R.A., Ghiran, A.M., Döller, V., Karagiannis, D.: Conceptual modeling education as a "design problem". Complex Syst. Inf. Model. Q. **21**, 21–33 (2019). https://doi.org/10.7250/csimq.2019-21.02
20. Ghiran, A.M., Osman, C.C., Buchmann, R.A.: Advancing conceptual modeling education towards a generalized model value proposition. In: Advances in Information Systems Development, LNISO 39, pp. 1–18. Springer, Berlin (2020). https://doi.org/10.1007/978-3-030-49644-9_1
21. Babeș-Bolyai University, FSEGA: UBB-FSEGA OMiLAB node—open models for business informatics. https://econ.ubbcluj.ro/omilab/publications.php (2023)
22. Karagiannis, D., Buchmann, R.A., Utz, W.: The OMiLAB digital innovation environment: agile conceptual models to bridge business value with digital and physical twins for product-service systems development. Comput. Ind. **138**, 103631 (2022). https://doi.org/10.1016/j.compind.2022.103631
23. Ontotext: Querying OpenAI GPT models. https://graphdb.ontotext.com/documentation/10.5/gpt-queries.html (2023)
24. Kahneman, D.: Thinking, Fast and Slow. Farrar, Straus and Giroux. ISBN 978-0374533557 (2013)
25. Karagiannis, D.: Conceptual modelling methods: the AMME agile engineering approach. In: Domain-Specific Conceptual Modeling - Concepts, Methods and ADOxx Tools, pp. 3–21. Springer (2022). https://doi.org/10.1007/978-3-030-93547-4_1
26. Floruț, C., Buchmann, R.A.: Semantic bridging between conceptual modeling standards and agile software projects conceptualizations. In: Proceedings of ISD 2022. AIS eLibrary. https://aisel.aisnet.org/isd2014/proceedings2022/managingdevops/7/ (2022)
27. Chiș, A., Ghiran, A.M.: Embracing conceptual modelling to enable the mediation role of enterprise information systems. In: Proceedings of ECIS 2023. AIS eLibrary. https://aisel.aisnet.org/ecis2023_rp/387/ (2023)
28. Uifălean, Ș.: Employing knowledge graphs for capturing semantic aspects of robotic process automation. In: Proceedings of CAISE 2023 Workshops, LNBIP 482, pp. 152–162. Springer (2023). https://doi.org/10.1007/978-3-031-34985-0_16
29. Cinpoeru, M.: Dereferencing service for navigating enterprise knowledge structures from diagrammatic representations. In: Proceedings of BIS 2017 Workshops, LNBIP 303, pp. 85–96. Springer (2017). https://doi.org/10.1007/978-3-319-69023-0_9

30. Miron, E.T., Muck, C., Karagiannis, D.: Transforming haptic storyboards into diagrammatic models: the Scene2Model tool. In: Proceedings of HICSS 2019. University of Hawaii. https://scholarspace.manoa.hawaii.edu/handle/10125/59494 (2019)
31. Moody, D.: The physics of notations: towards a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. **35**(6), 756–779 (2009). https://doi.org/10.1109/TSE.2009.67
32. isSemantic: RDF graph visualizer. https://issemantic.net/rdf-visualizer (2023)
33. IMCS University of Latvia: The OWLGrEd Ontology Editor. http://owlgred.lumii.lv/ (2023)
34. Buchmann, R.A., Karagiannis, D.: Enriching linked data with semantics from domain-specific diagrammatic models. Bus. Inf. Syst. Eng. **58**(5), 341–353 (2016). https://doi.org/10.1007/s12599-016-0445-1
35. Okoye, K., Tawil, A.R.H., Naeem, U., Lamine, E.: Semantic process mining towards discovery and enhancement of learning model analysis. In: Proceedings of IEEE 17th International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security, and IEEE 12th International Conference on Embedded Software and Systems, pp. 363–370. IEEE CS (2015). https://doi.org/10.1109/HPCC-CSS-ICESS.2015.164
36. Velitchkov, I., Anadiotis, G.: Personal Knowledge Graphs. Exapt Press (2023)
37. Karagiannis, D., Burzynski, P., Utz, W., Buchmann, R.: A metamodeling approach to support the engineering of modeling method requirements. In: Proceedings of RE 2019, pp. 199–210. IEEE (2019). https://doi.org/10.1109/RE.2019.00030
38. Buchmann, R.A.: The Purpose-Specificity Framework for Domain-Specific Conceptual Modeling. Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools, pp. 67–92. Springer (2022). https://doi.org/10.1007/978-3-030-93547-4_4
39. Laurenzi, E., Hinkelmann, K., van der Merwe, A.: An agile and ontology-aided modeling environment. In: Proceedings of PoEM 2018, LNBIP 335, pp. 221–237. Springer (2018). https://doi.org/10.1007/978-3-030-02302-7_14
40. Kühne, T., Jeusfeld, M.: Sanity-checking multiple levels of classification. In: Proceedings of ER 2023, LNCS 14320, pp. 162–180. Springer (2023). https://doi.org/10.1007/978-3-031-47262-6_9
41. Cabot, J.: Modeling should be an independent scientific discipline. Softw. Syst. Model. **21**, 2101–2107 (2022). https://doi.org/10.1007/s10270-022-01035-8
42. Baskerville, R.L., Myers, M.D., Yoo, Y.: Digital first: the ontological reversal and new challenges for information systems research. MIS Q. **44**(2), 509–523 (2020) https://aisel.aisnet.org/misq/vol44/iss2/3/
43. Recker, J., Lukyanenko, R., Jabbari, M., Samuel, B.M., Castellanos, A.: From representation to mediation: a new agenda for conceptual modeling research in a digital world. MIS Q. **45**(1), 269–300 (2021) https://aisel.aisnet.org/misq/vol45/iss1/10/
44. Voelz, A., Muck, C., Amlashi, D.M., Karagiannis, D.: Citizen-centric design of consumable services for smart cities. Digit. Gov. Res. Pract. **4**(3), 1–18 (2023). https://doi.org/10.1145/3597420
45. Bock, A., Frank, U.: Low-code platform. Bus. Inf. Syst. Eng. **63**, 733–740 (2021). https://doi.org/10.1007/s12599-021-00726-8
46. Solid: The solid project. https://solidproject.org/ (2023)
47. Fill, H.G., Fettke, P., Köpke, J.: Conceptual modeling and large language models: impressions from first experiments with ChatGPT. EMISAJ. **18**, 1–15 (2023). https://doi.org/10.18417/emisa.18.3
48. Clariso, R., Cabot, J.: Model-driven prompt engineering. In: Proceedings of MODELS 2023, pp. 47–54. ACM/IEEE CS (2023). https://doi.org/10.1109/MODELS58315.2023.00020
49. Nonaka, I., von Krogh, G.: Tacit knowledge and knowledge conversion: controversy and advancement in organisational knowledge creation theory. Organ. Sci. **20**, 635–652 (2009). https://doi.org/10.1287/orsc.1080.0412
50. OMiLAB NPO: KG4SDSE workshop series. https://www.omilab.org/activities/events/caise2024_kg4sdse/ (2023)

Spatial Conceptual Modeling: Anchoring Knowledge in the Real World

Hans-Georg Fill

Abstract This paper introduces the concept of spatial conceptual modeling, which allows anchoring mental world knowledge in the physical world using augmented reality technologies. For a first formal characterization, we describe a mapping from the spatial information concepts location, field, object, network, and event, as used in spatial computing, to conceptual modeling concepts using the FDMM formalism. This allows to identify necessary adaptations at the metamodeling level to make the approach applicable to arbitrary types of spatial conceptual modeling languages. Finally, possible application areas of spatial conceptual modeling in the medical domain, manufacturing and engineering, physical IT architectures and smart homes, supply chain management and logistics, civil engineering, and smart cities and cultural heritage are discussed.

Keywords Conceptual modeling · Spatial computing · Augmented reality · Knowledge representation

1 Introduction

Conceptual modeling is a widely-used technique for representing aspects about the physical and social world in order to support human understanding and communication [23, 46]. Further, conceptual models can act as interfaces to digital technologies, thereby using the contained knowledge for the configuration of machines and leading to the realization of innovative IT applications [13]. In recent years, there has been a growing interest in virtual reality, augmented reality, and mixed reality technologies [51, 63], which are part of the broader field of *spatial computing*. These technologies allow users to be fully immersed in virtual 3D environments through the use of special headsets (*virtual reality*), or to project virtual information onto

H.-G. Fill (✉)
University of Fribourg, Fribourg, Switzerland
e-mail: hans-georg.fill@unifr.ch

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_3

objects in the real world using see-through displays or smartphones (*augmented reality*), or combinations thereof (*mixed reality*).

The basic components of such applications have been available for a considerable period of time, and many of the technical concepts required have been well researched and developed to a high degree of maturity [57]. However, technological advances in the hardware of headsets and mobile devices have dramatically simplified the usability of such devices. In addition, the cost of these devices has come down significantly. Today, wireless headsets such as the Microsoft HoloLens,[1] the Meta Quest headsets[2] or, in the near future, the Apple Vision Pro[3] can be used out of the box, without having to connect them to powerful graphics workstations anymore. At the same time, the quality of headsets has improved over the years, affecting usability by reducing the potential for motion sickness and improving user acceptance [5].

As a result, it has become increasingly feasible to explore the use of these technologies in many business areas. These range today from applications in engineering, e.g. to support the assembly of machines [52], medical applications [45], robot interactions [10], IT management [7] to the gaming industry. While the technologies to compute and display the visual representations are an essential part of such applications, additional components are required to deliver the expected experience.

The field of spatial computing studies these aspects on a more general level [58]. According to a broad definition given by Greenwold, spatial computing can be characterized as "*human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces*" [22]. A central concept here is that such systems are aware of their location, be it in absolute terms such as position on the Earth, or in relative terms in the form of the distance to a reference point or origin. At a more granular level, location also includes orientation in space and how this is used to interact with a user. For example, in augmented reality applications, the user's position and orientation in space are used to compute overlays of the real world in the form of graphical information to augment the user's perception.

In the following, a synthesis of conceptual modeling with spatial computing will be described. This will be denoted as *spatial conceptual modeling*. In contrast to previous techniques and tools in conceptual modeling that relied on paper-based, two or three-dimensional electronic formats, spatial conceptual modeling will permit to anchor the contents of conceptual models in the real world. This allows to attach the knowledge in these models to physical objects and/or position it spatially. For accomplishing this, augmented reality technologies can be employed to visualize these anchorings if necessary. It will be discussed, which changes this requires on the level of metamodeling, i.e. the conceptualization and technical

[1] https://www.microsoft.com/en-us/hololens.

[2] https://www.meta.com/quest/.

[3] https://www.apple.com/apple-vision-pro/.

implementation of modeling languages and methods. Further, use cases for spatial conceptual modeling will be illustrated.

The remainder of this chapter is structured as follows. In Sect. 2, foundations on conceptual modeling and metamodeling, as well as spatial computing and augmented reality will be presented. Subsequently, in Sect. 3, the concept of spatial conceptual modeling will be elaborated and formally characterized using the FDMM formalism. In addition, potential use cases for spatial conceptual modeling will be illustrated. In Sect. 4 a conclusion and outlook on further research will be given.

2 Foundations

For achieving a common understanding of the components of spatial conceptual modeling, a brief overview on the foundations of conceptual modeling and metamodeling, as well as spatial computing and augmented reality will be outlined in the following.

2.1 Conceptual Modeling

Conceptual modeling is concerned with the explicit representation of some aspects of the physical or social world around us [46]. It is based on pre-defined elements or scripts that constrain what can be expressed in the models. In the sub-field of enterprise modeling for example, procedural knowledge about business processes, organizational knowledge, as well as knowledge about the enterprise architecture can be represented in this way [25, 32, 55]. This knowledge is typically codified using formal or semi-formal enterprise modeling languages [3]. In the past, a large range of frameworks and modeling methods have been developed for enterprise modeling, including for example the Business Process Management Systems (BPMS) Paradigm [31], Multi-Perspective Enterprise Modeling (MEMO) [21], the Semantic Object Model (SOM) [11], or the 4EM Method [56].

Besides these academic approaches, some of which have been successfully deployed in industry, also a range of industrial approaches were proposed in conceptual modeling. This includes for example a large number of international standards, which lead to benefits for companies in terms of compatibility and repeatability [39]. Examples for such standards include ArchiMate [61], BPMN [47], UML [48], or DMN [49], which can be either used individually or in combination, e.g. [8].

2.2 Metamodeling

Whereas conceptual models may also be created using pen and paper, any serious practical application requires today the use of IT-based modeling tools due to the complexity and size of the models. These tools not only permit to graphically represent the models but also to process them using algorithms and exchange them with third parties. Although such modeling tools may be created for one particular modeling language only, where the modeling primitives are hard-coded, the continuous evolution of modeling standards and languages would lead to much effort in their adaptation. In addition, conceptual modeling languages can be customized for specific purposes, or entirely new domain-specific languages can be developed to ensure optimal coverage of domain and user requirements [33, 34].

For these reasons, so-called metamodeling-based approaches have been designed. These correspond to typical approaches in knowledge representation and knowledge-based systems [28, 46, 60], where a *metamodel* acts as a terminological component (TBox), resulting from an iterative knowledge acquisition effort [29]. The metamodel thus formally defines the modeling language in such a way that it can be easily adapted if needed. This is in contrast to traditional approaches in compiler construction and language processing where the lexical and syntactic analysis as well as the actual code generation have to be explicitly specified [62]. In metamodel-based approaches, common abstractions of typical metamodels are provided which are used to define an individual metamodel. These are denoted as the *meta-metamodel*. These abstractions act as axioms and include for example concepts such as classes, relationclasses, attributes, or diagram types. When creating a metamodel, it is being reverted to these axioms for defining the terminological component. Metamodeling platforms—such as ADOxx for example [14]—can then interpret the metamodel based on the axioms, whose semantics are hard-coded in the platform [30]. In this way, the platforms can generate model editors for the specified modeling language. The created model instances then act as the assertion component (ABox) which contain the actual knowledge to be represented.

The main advantage of using metamodeling-based approaches and metamodeling platforms is the increased productivity in developing metamodels and thus new conceptual modeling languages. This is due to the pre-implemented concept interpretation functionality via the meta-metamodel in the terminology component. It eliminates the need for re-implementation and simplifies the creation of corresponding model editors and model processing environments. [28, 29].

2.3 Spatial Computing

Although a lot of research in *spatial computing* has been done mainly in the area of geographic information systems, it can also be viewed from a broader perspective [58]. According to Kuhn and Ballatore, spatial computing can be characterized by *spatial information* and *spatial computations* [36]. In the following these will be briefly summarized. For further details it is being referred to the original source [36].

Kuhn and Ballatore further classify the properties of spatial information into the five core content concepts *Location*, *Field*, *Object*, *Network*, and *Event* and two core quality concepts *Granularity* and *Accuracy*. Thereby, location as the most fundamental concept is regarded as a *relation*. The location is therefore always determined relative to something else, e.g. in a coordinate system relative to an origin. The field concept, which originates from physics permits to describe phenomena in a space of interest by a single value of an attribute. This is done by mathematical functions that map positions in space to values. An example would be a temperature field where each position in space is assigned a temperature value through a mathematical function. The object concept captures individual things that extend in space, including physical, mental, or social entities. Objects have an identity for tracking their properties and relations over time. The network concept is used to establish connections between objects. Networks thereby correspond to mathematical graphs, i.e. including nodes and edges, which may thus be used for computations, e.g. to determine the shortest path between two objects. The event concept refers to the temporal aspect in spatial information. It can have relations to fields, objects, and networks. The quality concept of granularity relates to the level of detail or precision, which is used for expressing spatial information, while accuracy refers to whether something is described correctly given a particular granularity.

In addition to the concepts for spatial information, Kuhn and Ballatore provide primitives for spatial computing operations, which can be combined for more complex computations. These are attached to spatial information content concepts and include for example topological operations such as *isAt* and *isIn* for locations to determine whether something is in contact *with* or contained *in* some other entity, as well as temporal operations such as *when* to determine the date of an event.

2.4 Augmented Reality

Augmented reality (AR) is a technique for superimposing virtual information such as visual media, audio, or haptic feedback on the physical environment in three dimensions that can be interacted with [2, 57]. This is achieved by using devices such as special headsets with see-through displays, headsets with pass-through cameras that display the real environment together with the virtual information on screens, as well as smartphones and tablets. In addition, special interaction devices may be used for haptic feedback or device-less interaction can be employed by using gestures. On a technical level, AR bridges the gap between the virtual and the real world—spatially and cognitively [57]. For realizing software applications that use augmented reality, a large number of concepts need to be understood and mastered on a technical level. These include for example *detectables* for guiding computer vision systems to identify real-world objects, *augmentations* that stand for virtual content that is fueled into the AR environment, *anchors* for specifying the positions of augmentations by connecting them to detectables, or *coordinate* and *transform* concepts for positioning objects in three-dimensional space [42]. Further, context information may need to be added to the AR application, e.g. to determine the current location of a user and determine which graphical representations should be presented [41, 59].

3 Spatial Conceptual Modeling

With the clarification of the terms outlined in the previous section, we can now advance to the description of the concept of *spatial conceptual modeling*. Several approaches have been described in the past for combining conceptual modeling and virtual and/or augmented reality. A recent literature survey of around 200 publications revealed that especially the fields of business process management and data modeling have investigated such combinations from various perspectives, including for example the elicitation and visualization of process, data, or enterprise architecture models using augmented and virtual reality, or the three-dimensional representation of models [43]. Further approaches that were found in the survey included approaches for the model-driven engineering of virtual reality and augmented reality applications, e.g. most recently [40, 42].

What seems to be missing so far, however, is a fundamental approach for combining the knowledge contained in conceptual models with entities in the real world on the level of metamodeling. Such an approach would permit to leverage arbitrary types of conceptual models in different languages to the spatial dimension. Although this could be done purely in formal mathematical notation, the practical benefits seem to lie in a combination with augmented reality technologies. These technologies permit to literally 'grasp' the knowledge of conceptual models in the real world.

We thus view augmented reality as a so-called *tertium comparationis* between the mental and the physical world. The mental world relates thereby to the knowledge as expressed through conceptual models and the physical world to all entities and things sensed in the real world. Following Čyras and Lachmayer, the tertium comparationis refers to the "quality that two things that are being compared have in common" [9][p.15]. As depicted in Fig. 1, the commonalities of the mental and

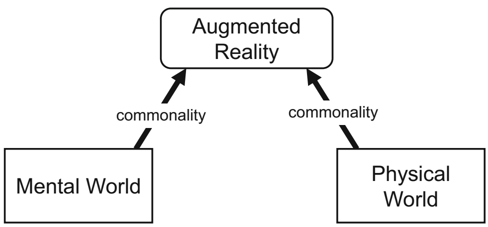


Fig. 1 Augmented reality as the tertium comparationis between the mental and the physical world

the physical world are thereby mediated via augmented reality, for example by superimposing knowledge on how to operate a machine that is part of the mental world and made explicit via a conceptual model on the actual buttons and switches of the physical machine.

Augmented reality thereby enables the anchoring of information stemming from conceptual models to entities in the real world by providing the necessary technical concepts. In terms of spatial information content it may however be needed to provide additional information, which is typically not found so far in conceptual models. This includes for example information on the location of information objects in physical space, as well as different levels of granularity in terms of different levels of detail required for the expression of knowledge in conceptual models and on the physical level—see for an example the work by Crevoiserat et al. [7].

In a fictitious example, conceptual models could be anchored in the real world as follows. The models could contain information on the enterprise architecture of a company which is displayed in the server room when people with AR headsets enter the room. Thus, it would not be necessary anymore to operate a laptop or tablet and search for the right model. Rather, the model could be displayed once the people look at a particular server rack, e.g. to determine which applications run on that server and who is responsible for it.

We can further identify different levels of anchoring conceptual models in the real world. As shown in Table 1 five levels are proposed. First, level 0 stands for the classical, 2D-based modeling without any spatialization. Subsequently, Level 1 '*Unanchored Spatial CM*' characterizes approaches where models are just presented in space, e.g. by displaying them on an AR headset or narrating their content on audio devices in space. However, on this level models are not tied to a particular physical location. This may be useful for automatically presenting models to users upon sensing and reasoning about their behavior, independent of their location. Level 2 '*Model-Anchored Spatial CM*' stands for the anchoring of models to points of interest in space. For example, when a user approaches a machine at a defined physical location, a particular model with operating instructions is shown.

Level 3 '*Statically Anchored Spatial CM*' goes one step further and refers to the anchoring of individual model elements in space. Instead of placing the whole model at some point in space, now single elements of a model can be tied to physical locations. For example, steps in the operation process of a machine may be anchored

Table 1 Anchoring levels of spatial conceptual modeling (CM)

Level	Designation	Description of anchoring level
Level 0	Traditional CM	No spatialization, classical 2D modeling
Level 1	Unanchored spatial CM	Spatialization of models without anchoring
Level 2	Model-anchored spatial CM	Anchoring of models to points of interest in space
Level 3	Statically anchored spatial CM	Static anchoring of model elements in space
Level 4	Dynamically anchored spatial CM	Dynamic anchoring of model elements in space

to the various switches of that machine to indicate which switch needs to be turned next. Level 4 '*Dynamically Anchored Spatial CM*' extends this anchoring of model elements then in a dynamic manner. Instead of statically anchoring model elements, this may be done dynamically, e.g. based on some reasoning about the current context [41]. For example, a workflow for operating switches on different machines may be dynamically adapted based on some user action.

3.1 Formal Characterization of Spatial Conceptual Modeling

For realizing spatial conceptual modeling we can derive in the next step how spatial information concepts can be represented in conceptual modeling on a metamodeling level. For this purpose we will revert to the constructs of the FDMM formalism [17, 18]. FDMM has been designed to formally describe metamodels and their model instances on a technology-independent level. In contrast to other formalisms it strives for ease-of-use and simplicity in the mathematical formulation, so that also people with only little background in formal specifications can understand and apply it.

FDMM defines metamodels $\mathbf{MM}$ as a tuple of the form $\mathbf{MM} = \langle \mathbf{MT}, \preceq, \text{domain}, \text{range}, \text{card} \rangle$, where $\mathbf{MT}$ stands for a set of model types. Each model type $\mathbf{MT}_i$ has in turn a tuple of object types $\mathbf{O}_i^T$, data types $\mathbf{D}_i^T$, and attributes $\mathbf{A}_i$, i.e. $\mathbf{MT}_i = \langle \mathbf{O}_i^T, \mathbf{D}_i^T, \mathbf{A}_i \rangle$. Object types are used to both represent the types of nodes and edges in typical model diagrams or as template for arbitrary objects. Attributes can be attached to object types via the domain function. The range function determines the type of content of an attribute type—including data types or other object or model types—and the card function specifies the cardinality of attribute values in model instances. $\preceq$ is an ordering of object types for specifying inheritance relationships between object types. For the scope of this paper we omit the detailed formal relationships between the constructs of metamodels as well as the instantiation part of FDMM and refer interested readers to [17, 18] as well as further applications of FDMM in [19, 26, 27].

In the following we will show how the information concepts from spatial computing can be mapped to metamodeling based on the outline in Sect. 2.3 following Kuhn and Ballatore [36]. These mappings are shown in Table 2. We first consider the respective concept from spatial computing and then the corresponding FDMM concepts for spatial conceptual modeling. Finally a brief description is added.

Starting with the concept of *Location*, this requires on the side of conceptual modeling that all modeling objects can be anchored in three-dimensional space using *coordinates*. However, due to the multitude of coordinate systems necessary for spatial computing applications—e.g. GPS coordinates, coordinates in Building Information Modeling (BIM), coordinates of graphical devices and sensors, etc.—we subsume these under the attribute set $\mathbf{A}_{\text{coord}}$. This set is attached to any object

Table 2 Mappings of spatial information concepts to FDMM concepts for spatial conceptual modeling

Spatial information concept	FDMM concept	Description
Location	$\mathbf{A}_{\text{coord}}$	Set of coordinate attributes
	$\text{domain}(\mathbf{A}_{\text{coord}}) = \mathbf{O}^T$	Every object type has coordinate attributes
	$\mathbf{A}_{\text{transform}}$	Set of transform attributes
	$\text{domain}(\mathbf{A}_{\text{transform}}) = \mathbf{O}^T$	Every object type has transform attributes
Field	$\mathbf{O}^T_{\text{field}}, \mathbf{A}_{\text{field}}$	Sets of field object types and attributes
	$\text{domain}(\mathbf{A}_{\text{field}}) = \mathbf{O}^T_{\text{field}}$	Field attributes are assigned to field object types
Object	$\mathbf{a}_{\text{uuid}}$	UUID attribute for object identity
	$\text{domain}(\mathbf{a}_{\text{uuid}}) = \mathbf{O}^T$	Every object type has a UUID attribute
	$\mathbf{O}^T_{\text{real}} \subseteq \mathbf{O}^T$	Real object types refer to objects in the real world
	$\mathbf{O}^T_{\text{virt}} \subseteq \mathbf{O}^T$	Virtual object types stand for virtual objects
	$(\mathbf{O}^T_{\text{real}} \cup \mathbf{O}^T_{\text{virt}}) \subset \mathbf{O}^T$	The set of object types comprises all real and virtual object types
	$\mathbf{O}^T_{\text{rv}} \subseteq \mathbf{O}^T$	Object types for relating real and virtual object types
	$\mathbf{A}_{\text{vizrep}}$	Set of visualization representation attributes
	$\text{domain}(\mathbf{A}_{\text{vizrep}}) = \mathbf{O}^T$	Every object type has a set of visualization representation attributes
Network	$\mathbf{O}^T$	Nodes and edges represented as FDMM object types
Event	$\mathbf{O}^T_{\text{event}}$	Event object types for temporal events
	$\mathbf{A}_{\text{event}}$	Attributes for events for expressing temporal properties
	$\mathbf{O}^T_{\text{Temp}}$	Temporal event object types for temporal relations between events
	$\mathbf{O}^T_{\text{part}}$	Participation object types for relations between events and other object types

type using a domain function. Similarly, objects may have an orientation in space that needs to be considered and that we regard also under location. Therefore, we add a set $\mathbf{A}_{\text{transform}}$ that holds this information. Note that also for coordinate transforms many variants exist such as rotation matrices, Euler angles, quaternions and so on.

The concept of a *Field* can also be represented in conceptual modeling via object types and assigned attributes. We denote these through $\mathbf{O}^T_{\text{field}}$, $\mathbf{A}_{\text{field}}$. In this way, for example, a temperature field could be created through the instantiation of a field object type and values from attributes assigned to it that hold e.g. information about the mathematical function specifying the physics of the field. Further, such a field can be combined and extended with references to other objects types as done in conceptual modeling for adding semantic information, e.g. to represent user actions that are affected by the temperature field in a laboratory environment.

For the *Object* concept from spatial computing, the identity is a key property. Therefore, we foresee a distinct attribute $\mathbf{a}_{\text{uuid}}$ that is used to attach a universally unique identifier to each object type. This permits the decentralized creation of new, uniquely identifiable objects across an arbitrary number of applications. We further divide the overall set of object types into object types referring to the real world $\mathbf{O}^T_{\text{real}}$ and those standing for virtual objects $\mathbf{O}^T_{\text{virt}}$. This can be used for example in augmented reality applications to hold information about objects in the real world, e.g. in the form of markers as surrogates for these objects or for identifying the position and pose of real objects using object detection techniques [64]. Virtual object types are used to represent the traditional conceptual modeling elements and relations, e.g. a place in a Petri net diagram, a class in UML, an inheritance relationship type in UML—or, more advanced types such as augmentations for augmented reality applications. In addition, object types for relating real and virtual object types are foreseen, which are denoted as $\mathbf{O}^T_{\text{rv}}$. These permit to establish connections between virtual and real world objects, e.g. for anchoring an augmentation with a real world object. The attributes, domain, and range functions for referencing the related real or virtual object types are omitted here for brevity. An essential aspect of conceptual models is their graphical representation, even more so in three-dimensional space. Therefore, the set $\mathbf{A}_{\text{vizrep}}$ is foreseen to include all necessary attributes related to the visual representation of object types. This may include static representations in two, three or more dimensions—e.g. including time and animation aspects—as well as dynamic aspects, e.g. for the realization of dynamic state changes in visualizations based on attribute states [12].

The *Network* concept of spatial computing largely corresponds to the typical constructs in conceptual models for representing mathematical graphs in various forms. Therefore, we can directly map it to FDMM object types, which are used to represent nodes and edges in graphs.

The *Event* concept is used to introduce temporal aspects in spatial computing. We foresee a set of object types $\mathbf{O}^T_{\text{event}}$ for the events themselves, a set $\mathbf{A}_{\text{event}}$ for representing temporal properties of events in the form of attributes such as durations and time units, the object type $\mathbf{O}^T_{\text{Temp}}$ for temporal relations between events such as before, after, etc., and the object types $\mathbf{O}^T_{\text{part}}$ for participating relationships between events and any other object type.

Concerning the operations for spatial computing, we omit those at this stage due to the fact that FDMM does not foresee fundamental operations for models at the moment. A possible formalization could be based for example on topological relations for representing knowledge [24], as well as approaches for qualitative spatial reasoning [6]. This would need to be mapped to fundamental operations on model instances such as iterations over model elements, creation, modification and deletion of elements, or operations affecting the user interface of model editors such as markings or animations.

3.2 Possible Applications Areas of Spatial Conceptual Modeling

From a general perspective, spatial conceptual modeling seems most adequate for applications that combine knowledge-intensive areas with physical interactions. In such cases, knowledge needs to be elicited in the form of conceptual models, which can then either be mapped directly or via intermediate representations to the physical environment. In the following we list some potential application areas.

Healthcare and Medical Domain In this domain, conceptual modeling has been used for example to represent clinical pathways, treatment processes, or regulatory processes, e.g. [4, 16]. The integration of spatial aspects could help to support medical education about treatments or personnel in care delivery, where augmented reality has already been successfully investigated [45].

Engineering and Manufacturing Several approaches have been developed for using augmented reality in this field, e.g. in maintenance, collaborative design, layouting, or training [52]. The addition of conceptual modeling could help to deal with the challenge of integrating the large amounts of data and knowledge necessary for such applications and for making the working of AR applications more transparent. Further, spatial conceptual modeling could complement digital twin approaches in engineering by adding knowledge and process aspects to the traditionally used CAD models [1].

Physical IT Architectures and Smart Homes In IT and enterprise architecture scenarios, approaches have been developed to visualize systems using augmented reality [53]. Whereas also such approaches may benefit from spatial conceptual modeling through directly retrieving the necessary data from the models, we see further potential in the combination with physical IT architectures, e.g. to support maintenance, for the physical wiring and optimization of communication networks, or for end users in smart home scenarios [37], where the combination with conceptual models could already be successfully demonstrated [7].

Supply Chain Management and Logistics For this domain, the application of augmented reality has been investigated for the optimization of business processes, e.g. for supporting the selection of next steps, for picking items, or for monitoring processes [50, 54]. However, as derived by Rejeb et al. in a recent literature survey, several technical, organizational, and ergonomical challenges persist in this area [54]. For this purpose, spatial conceptual modeling could help to reduce the complexity of integrating augmented reality technologies in such processes and increase the transparency of according applications. This could support both the acceptance of the technologies by the involved workers—e.g. for addressing privacy issues and fear of control—as well as to ease the alignment of the technology with organizational requirements.

Civil Engineering, Smart Cities, and Cultural Heritage At the core of these areas are today large amounts of information that need to be processed electronically. May it be Building Information Models (BIM) that are used for the engineering and operation of buildings, data for environmental optimizations and sustainability, legal and compliance aspects, or historical knowledge about cultural artifacts. The use of spatial conceptual modeling could aid here in aligning the vast amounts of knowledge with the diverse sources of spatial data and presenting them through augmented reality applications [35, 38, 44]. Examples would include the integration of knowledge about cultural heritage and spatial architectural data, as well as data for environmental and sustainability purposes, e.g. in smart city planning or for informing citizens about environmental-friendly behavior in the built environment.

In summary, the provision of a generic approach for spatial conceptual modeling that is applicable to diverse application areas would not only permit the easier exchange of knowledge on implementing augmented reality applications. It would also support these areas with a common technical approach for integrating data and knowledge, thus potentially making the implementation more efficient and effective.

4 Conclusion and Outlook

In this paper, a first outline of the approach of spatial conceptual modeling has been presented, including an intimal formal characterization. At its core, spatial conceptual modeling aims to bridge the gap between the mental and physical world through augmented reality applications in terms of knowledge aspects. There are however a number of open issues that will need to be addressed. This concerns both conceptual and implementation aspects of spatial conceptual modeling. For example, it will need to be investigated, whether existing metamodeling platforms can be used for its realization in terms of required concepts and technologies [30]. Further, it will need to be evaluated whether the wide range of existing conceptual modeling approaches is adequate for transitioning them to the spatial environment or which changes are required [34]. Finally, upcoming technologies in artificial intelligence could be used for easing the creation of spatial representations of model elements, which requires otherwise specialized knowledge about three dimensional modeling [15, 20]. Last but not least, business models and further use cases for spatial conceptual modeling need to be designed and evaluated, e.g. in the context of metaverse-like environments.

References

1. Albuquerque, G., Fischer, P.M., Azeem, S.M., Bernstein, A.C., Utzig, S., Gerndt, A.: Digital twins as foundation for augmented reality applications in aerospace. In: Springer Handbook of Augmented Reality, pp. 881–900. Springer, Berlin (2023)
2. Azuma, R.T.: A survey of augmented reality. Presence Teleoperators Virtual Environ. **6**(4), 355–385 (1997)
3. Bork, D., Fill, H.: Formal aspects of enterprise modeling methods: a comparison framework. In: 47th Hawaii International Conference on System Sciences, HICSS 2014, pp. 3400–3409. IEEE Computer Society (2014). https://doi.org/10.1109/HICSS.2014.422
4. Braun, R., Schlieter, H., Burwitz, M., Esswein, W.: Extending a business process modeling language for domain-specific adaptation in healthcare. In: Thomas, O., Teuteberg, F. (eds.) Smart Enterprise Engineering: 12. Internationale Tagung Wirtschaftsinformatik, WI 2015, Osnabrück, Germany, March 4-6, 2015, pp. 468–481 (2015). http://www.wi2015.uni-osnabrueck.de/Files/WI2015-D-14-00282.pdf
5. Caserman, P., Garcia-Agundez, A., Gámez Zerban, A., Göbel, S.: Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook. Virtual Real. **25**(4), 1153–1170 (2021)
6. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundam. Inform. **46**(1–2), 1–29 (2001)
7. Crevoiserat, S., Muff, F., Fill, H.: Towards augmented reality applications for IT maintenance tasks based on ArchiMate models. In: Proceedings of the ER Demos and Posters 2023 co-located with International Conference on Conceptual Modeling (ER 2023). CEUR-WS (2023). https://ceur-ws.org/Vol-3618/pd_paper_5.pdf
8. Curty, S., Fill, H.: Exploring the systematic design of blockchain-based applications using integrated modeling standards. In: Bork, D., Barat, S., Asprion, P.M., Marcelletti, A., Morichetta, A., Schneider, B., Kulkarni, V., Breu, R., Zech, P. (eds.) Proceedings of the PoEM 2022 Workshops and Models at Work co-located with Practice of Enterprise Modelling 2022, CEUR Workshop Proceedings, vol. 3298. CEUR-WS.org (2022). URL https://ceur-ws.org/Vol-3298/paper_BES_6037.pdf
9. Cyras, V., Lachmayer, F.: Essays on the Visualisation of Legal Informatics. Springer, Berlin (2023). https://doi.org/10.1007/978-3-031-27957-7
10. Delmerico, J.A., Poranne, R., Bogo, F., Oleynikova, H., Vollenweider, E., Coros, S., Nieto, J.I., Pollefeys, M.: Spatial computing and intuitive interaction: bringing mixed reality and robotics together. IEEE Robotics Autom. Mag. **29**(1), 45–57 (2022). https://doi.org/10.1109/MRA.2021.3138384
11. Ferstl, O.K., Sinz, E.J.: Objektmodellierung betrieblicher informationssysteme im semantischen objektmodell (SOM). Wirtschaftsinf. **32**(6), 566–581 (1990)
12. Fill, H.G.: Visualisation for Semantic Information Systems. Springer Science & Business Media, Berlin (2009)
13. Fill, H.: Enterprise modeling: from digital transformation to digital ubiquity. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, FedCSIS 2020. Annals of Computer Science and Information Systems, vol. 21, pp. 1–4 (2020). https://doi.org/10.15439/2020F001
14. Fill, H., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **8**(1), 4–25 (2013). https://doi.org/10.18417/EMISA.8.1.1
15. Fill, H.G., Muff, F.: Visualization in the era of artificial intelligence: experiments for creating structural visualizations by prompting large language models. arXiv (2023). https://doi.org/10.48550/ARXIV.2305.03380. https://arxiv.org/abs/2305.03380
16. Fill, H., Eberhart, A., Laslop, A., Reischl, I., Lang, T., Karagiannis, D.: An approach to support the performance management of public health authorities using an IT based modeling method. In: 10. Internationale Tagung Wirtschaftsinformatik, Zürich, 16.-18. February 2011, p. 115 (2011). http://aisel.aisnet.org/wi2011/115

17. Fill, H., Redmond, T., Karagiannis, D.: FDMM: a formalism for describing adoxx meta models and models. In: Maciaszek, L.A., Cuzzocrea, A., Cordeiro, J. (eds.) ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems, vol. 3, Wroclaw, Poland, 28 June - 1 July, 2012, pp. 133–144. SciTePress (2012)
18. Fill, H., Redmond, T., Karagiannis, D.: Formalizing meta models with FDMM: the adoxx case. In: Cordeiro, J., Maciaszek, L.A., Filipe, J. (eds.) Enterprise Information Systems - 14th International Conference, ICEIS 2012, Wroclaw, Poland, June 28 - July 1, 2012, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 141, pp. 429–451. Springer (2012). https://doi.org/10.1007/978-3-642-40654-6_26
19. Fill, H., Hickl, S., Karagiannis, D., Oberweis, A., Schoknecht, A.: A formal specification of the horus modeling language using FDMM. In: 11. Internationale Tagung Wirtschaftsinformatik, Leipzig, Germany, February 27 – March 1, 2013, p. 73 (2013). http://aisel.aisnet.org/wi2013/73
20. Fill, H., Fettke, P., Köpke, J.: Conceptual modeling and large language models: impressions from first experiments with ChatGPT. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **18**, 3 (2023). https://doi.org/10.18417/EMISA.18.3
21. Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. Softw. Syst. Model. **13**(3), 941–962 (2014). https://doi.org/10.1007/S10270-012-0273-9
22. Greenwold, S.: Spatial computing. Massachusetts Institute of Technology, Master Thesis, 2003
23. Härer, F., Fill, H.: Past trends and future prospects in conceptual modeling - A bibliometric analysis. In: Dobbie, G., Frank, U., Kappel, G., Liddle, S.W., Mayr, H.C. (eds.) Conceptual Modeling - 39th International Conference, ER 2020, Vienna, Austria, November 3–6, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12400, pp. 34–47. Springer (2020). https://doi.org/10.1007/978-3-030-62522-1_3
24. Hernández, D.: Qualitative Representation of Spatial Knowledge. Springer, Berlin (1994)
25. Hinkelmann, K., Gerber, A., Karagiannis, D., Thönssen, B., van der Merwe, A., Woitsch, R.: A new paradigm for the continuous alignment of business and IT: combining enterprise architecture modelling and enterprise ontology. Comput. Ind. **79**, 77–86 (2016). https://doi.org/10.1016/J.COMPIND.2015.07.009
26. Johannsen, F., Fill, H.: Supporting knowledge elicitation and analysis for business process improvement through a modeling tool. In: Thomas, O., Teuteberg, F. (eds.) Smart Enterprise Engineering: 12. Internationale Tagung Wirtschaftsinformatik, WI 2015, Osnabrück, Germany, March 4–6, 2015, pp. 752–766 (2015). http://www.wi2015.uni-osnabrueck.de/Files/WI2015-D-14-00093.pdf
27. Johannsen, F., Fill, H.: Meta modeling for business process improvement. Bus. Inf. Syst. Eng. **59**(4), 251–275 (2017). https://doi.org/10.1007/S12599-017-0477-1
28. Karagiannis, D.: Wissensbasierte Datenbanken, vol. 8.2. R. Oldenbourg Verlag, Munich (1994)
29. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics, pp. 5–10 (2015)
30. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) E-Commerce and Web Technologies, Third International Conference, EC-Web 2002, Aix-en-Provence, France, September 2–6, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2455, p. 182. Springer (2002). https://doi.org/10.1007/3-540-45705-4_19
31. Karagiannis, D., Junginger, S., Strobl, R.: Introduction to business process management systems concepts. In: Scholz-Reiter, B., Stickel, E. (eds.) Business Process Modelling, pp. 81–106. Springer, Heidelberg (1996)
32. Karagiannis, D., Buchmann, R.A., Bork, D.: Managing consistency in multi-view enterprise models: an approach based on semantic queries. In: 24th European Conference on Information Systems, ECIS 2016 (2016)
33. Karagiannis, D., Mayr, H.C., Mylopoulos, J.: Domain-specific conceptual modeling. Springer, Berlin (2016)

34. Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W.: Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools. Springer Nature, Berlin, (2022)
35. Kim, H., Matuszka, T., Kim, J.I., Kim, J., Woo, W.: Ontology-based mobile augmented reality in cultural heritage sites: information modeling and user study. Multimed. Tools Appl. **76**, 26001–26029 (2017)
36. Kuhn, W., Ballatore, A.: Designing a language for spatial computing. In: Bação, F., Santos, M.Y., Painho, M. (eds.) AGILE 2015 - Geographic Information Science as an Enabler of Smarter Cities and Communities, Lisboa, Portugal, 9-12 June 2015. Lecture Notes in Geoinformation and Cartography, pp. 309–326. Springer (2015). https://doi.org/10.1007/978-3-319-16787-9_18
37. Mahroo, A., Greci, L., Sacco, M.: Holohome: an augmented reality framework to manage the smart home. In: Augmented Reality, Virtual Reality, and Computer Graphics: 6th International Conference, AVR 2019, Santa Maria al Bagno, Italy, June 24–27, 2019, Proceedings, Part II 6, pp. 137–145. Springer (2019)
38. Maietti, F., Medici, M., Ferrari, F.: From semantic-aware digital models to augmented reality applications for architectural heritage conservation and restoration. Disegnarecon **14**(26), 17–1 (2021)
39. Moser, C., Buchmann, R.A., Utz, W., Karagiannis, D.: CE-SIB: a modelling method plug-in for managing standards in enterprise architectures. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) Conceptual Modeling - 36th International Conference, ER 2017, pp. 21–35. Springer (2017). https://doi.org/10.1007/978-3-319-69904-2_2
40. Muff, F., Fill, H.: Initial concepts for augmented and virtual reality-based enterprise modeling. In: Lukyanenko, R., Samuel, B.M., Sturm, A. (eds.) Proceedings of the ER Demos and Posters 2021 co-located with 40th International Conference on Conceptual Modeling (ER 2021), St. John's, NL, Canada, October 18-21, 2021. CEUR Workshop Proceedings, vol. 2958, pp. 49–54. CEUR-WS.org (2021). https://ceur-ws.org/Vol-2958/paper9.pdf
41. Muff, F., Fill, H.: A framework for context-dependent augmented reality applications using machine learning and ontological reasoning. In: Martin, A., Hinkelmann, K., Fill, H., Gerber, A., Lenat, D., Stolle, R., van Harmelen, F. (eds.) Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022), vol. 3121. CEUR-WS.org (2022). https://ceur-ws.org/Vol-3121/paper11.pdf
42. Muff, F., Fill, H.: A domain-specific visual modeling language for augmented reality applications using webxr. In: Almeida, J.P.A., Borbinha, J., Guizzardi, G., Link, S., Zdravkovic, J. (eds.) Conceptual Modeling - 42nd International Conference, ER 2023. Lecture Notes in Computer Science, vol. 14320, pp. 334–353. Springer (2023). https://doi.org/10.1007/978-3-031-47262-6_18
43. Muff, F., Fill, H.: Past achievements and future opportunities in combining conceptual modeling with VR/AR: a systematic derivation. In: Shishkov, B. (ed.) Business Modeling and Software Design - 13th International Symposium, BMSD 2023, pp. 129–144. Springer (2023). https://doi.org/10.1007/978-3-031-36757-1_8
44. Muff, F., Fill, H., Kahlig, E., Kahlig, W.: Kontextabhängige Rechtsvisualisierung mit Augmented Reality. HMD Prax. Wirtsch. **59**(1), 92–109 (2022). https://doi.org/10.1365/S40702-021-00832-X
45. Munzer, B.W., Khan, M.M., Shipman, B., Mahajan, P.: Augmented reality in emergency medicine: a scoping review. J. Med. Internet Res. **21**(4), e12368 (2019)
46. Mylopoulos, J.: Conceptual modelling and telos. Conceptual Modelling, Databases, and CASE: An Integrated View of Information System Development, pp. 49–68. Wiley Professional Computing (1992)
47. Object Management Group: Business Process Model and Notation (BPMN) V2.0.2. https://www.omg.org/spec/BPMN (2014)
48. Object Management Group: Unified Modeling Language V2.5.1. https://www.omg.org/spec/UML/2.5.1/ (2017)
49. Object Management Group: Decision Model and Notation V1.5 beta. https://www.omg.org/spec/DMN (2023)

50. Plakas, G., Ponis, S., Agalianos, K., Aretoulaki, E., Gayialis, S.: Augmented reality in manufacturing and logistics: Lessons learnt from a real-life industrial application. Procedia Manuf. **51**, 1629–1635 (2020)
51. PricewaterhouseCoopers (PwC): What does virtual reality and the metaverse mean for training? (2022). https://www.pwc.com/us/en/tech-effect/emerging-tech/virtual-reality-study.html. Last accessed: 02 Jan 2024
52. Regenbrecht, H., Baratoff, G., Wilke, W.: Augmented reality projects in the automotive and aerospace industries. IEEE Comput. Graph. Appl. **25**(6), 48–56 (2005)
53. Rehring, K., Greulich, M., Bredenfeld, L., Ahlemann, F.: Let's get in touch - decision making about enterprise architecture using 3d visualization in augmented reality. In: 52nd Hawaii International Conference on System Sciences, pp. 1–10 (2019). https://hdl.handle.net/10125/59617
54. Rejeb, A., Keogh, J.G., Leong, G.K., Treiblmaier, H.: Potentials and challenges of augmented reality smart glasses in logistics and supply chain management: a systematic literature review. Int. J. Prod. Res. **59**(12), 3747–3776 (2021)
55. Sandkuhl, K., Fill, H., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A.L., Schwabe, G., Uludag, Ö., Winter, R.: From expert discipline to common practice: a vision and research agenda for extending the reach of enterprise modeling. Bus. Inf. Syst. Eng. **60**(1), 69–80 (2018). https://doi.org/10.1007/S12599-017-0516-Y
56. Sandkuhl, K., Stirna, J., Persson, A., Wißotzki, M.: Enterprise Modeling - Tackling Business Challenges with the 4EM Method. The Enterprise Engineering Series. Springer, Berlin (2014). https://doi.org/10.1007/978-3-662-43725-4
57. Schmalstieg, D., Höllerer, T.: Augmented Reality: Principles and Practice. Addison-Wesley Professional, Boston (2016)
58. Shekhar, S., Feiner, S.K., Aref, W.G.: Spatial computing. Commun. ACM **59**(1), 72–81 (2016). https://doi.org/10.1145/2756547
59. Shishkov, B., Fill, H., Ivanova, K., van Sinderen, M., Verbraeck, A.: Incorporating trust into context-aware services. In: Shishkov, B. (ed.) Business Modeling and Software Design - 13th International Symposium, BMSD 2023, vol. 483, pp. 92–109. Springer (2023). https://doi.org/10.1007/978-3-031-36757-1_6
60. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data Knowl. Eng. **25**(1–2), 161–197 (1998)
61. The Open Group: Archimate® 3.2 specification. https://pubs.opengroup.org/architecture/archimate3-doc/ (2023)
62. Wirth, N.: Compiler Construction. Addison-Wesley, Boston (1996)
63. Wohlgenannt, I., Simons, A., Stieglitz, S.: Virtual reality. Bus. Inf. Syst. Eng. **62**(5), 455–461 (2020). https://doi.org/10.1007/S12599-020-00658-9
64. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. Proc. IEEE **111**(3), 257–276 (2023). https://doi.org/10.1109/JPROC.2023.3238524

Process vs. Capability: A Reflection from a Practitioner's Perspective

Anke Helmes

Abstract Within many large companies Enterprise Architecture Management (EAM) and Business Process Management (BPM) have established as management approaches over the last decades. While EAM pursues a holistic view on the elements of a company and their relations to each other, whereby "process" and "capability" are two main elements, BPM specializes in the design and optimization of processes. Although both approaches are mature and a lot of material has been published already, some questions come up in the practical context of managing processes and capabilities as well as their relation to each other. Based on the foundation about capability management as part of EAM and about BPM, the paper elaborates on two main challenges in the context of managing "capabilities" and "processes" within a company from a practitioner's perspective: (1) Questioned value add of having both a capability map and a process landscape, (2) Uncertainty on which levels capabilities and processes should be related to each other. Furthermore, for those observations a proposal is provided on how to address those challenges.

Keywords Enterprise architecture · Business process management · Capability management · Process · Capability · Process landscape · Capability map

1 Introduction

In many (European) companies, an Enterprise Architecture Management (EAM) as well as a Business Process Management (BPM) have been established over the last decades. This can be observed especially for large companies, but also for mid-sized companies irrespective of their industry.

While EAM pursues a holistic view on the elements of a company and their relations to each other, whereby "process" and "capability" are two main elements

A. Helmes (✉)
Munich, Germany

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_4

next to e.g. "application" or "technology", BPM specializes in the design and optimization of processes. Within BPM, processes are represented on a higher abstraction level in the form of so called process landscapes and on a lower level of detail in the form of business process flows. Within EAM, processes are represented in (hierarchical) lists or architectural models, whereas capabilities are visualized with the help of so called capability maps.

Although both management approaches are mature and a lot of conceptual and practical material has been published already, some questions come up in the practical context: What is the value add of having a capability map provided by EAM on the one hand and a process landscape provided by BPM on the other hand? On which levels of granularity can processes be linked to capabilities and/or vice versa?

The objective of this contribution is to further elaborate on those two questions and to provide some insights from a practitioners' perspective.

To achieve this, Sect. 2 will lay the foundation by describing the understanding of a "capability" and a "process" in the disciplines of Enterprise Architecture Management and Business Process Management. Based on this, Sect. 3 will go into more detail about the raised questions by elaborating on observations and challenges in the practical context of managing capabilities and processes. In Sect. 4, a proposal will be shared to address those questions from a practitioner's point of view. Finally, Sect. 5 provides a short summary of the contribution.

2 Understanding "Capability" and "Process" in the Context of Enterprise Architecture Management and Business Process Management

2.1 Capability Management as Part of Enterprise Architecture Management

Enterprise Architecture Management is a management approach which roots go back more than 20 years ago. Its main tasks are the design, maintenance and implementation of an *Enterprise Architecture [EA]* [1, p. 27]. Although a lot of definitions about an enterprise architecture exist, many of them go back to the definition in the IEEE standard 1471-2000 [2, p. 10]: "*Architecture: the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.*" Based on this, an enterprise architecture is understood as the organization of a system/company, i.e. its main elements and their relations to each other as well as their relations to the surrounding ecosystem [1, p. 27].

To describe an enterprise architecture often the standardized and well established modeling language ArchiMate® is used [3, p. 35, 37]. ArchiMate® does not only define and describe the different elements of an EA, but also the possible and recommended relations between those elements [3, p. 38]. One of these elements is a "*Capability*".

A capability is understood as a specific ability, which an organization possesses or wants to possess. A capability describes "What" a company can do or wants to do, but not "How", "Where" or "Why". Capabilities are often visualized with the help of *Capability Maps*. A capability map graphically visualizes the capabilities of an organization and their hierarchical structure. Theoretically multiple levels of detail can be used to structure capabilities. To give an example the capability "Customer Management" could be further decomposed into "Customer Data Management", "Customer Relations Management", "Customer Order Management", etc. [4]. Especially high-level capabilities are used by enterprise architects to strategically plan the further development of the organization. On the one hand, a capability map can be used as a communication instrument to align with senior management on the overall strategy/orientation of the organization. On the other hand capabilities are seen as the linking element between strategic objectives and their implementation by different transformation projects [5, p. 92 ff.].

Next to the "Capability" element, a "Process" is also a relevant element in the context of the business architecture layer within an enterprise architecture. For a "process" ArchiMate® differentiates various concepts [6]: "Value Stream" (and "Value Stream Stage") as well as "Business Process". Whereas a value stream "*represents a sequence of activities that create an overall result for a customer, stakeholder, or end user*" [6], a business process "*represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services within the context of a business capability instance*" [6]. Next to the elements themselves, the ArchiMate® standard also includes a definition, how these elements should be related to each other [6]: On the one hand a capability is enabled by a business process. On the other hand a capability enables a value stream (stage).

2.2 Business Process Management

Also *Business Process Management* is a management discipline with a long tradition. Over the last three decades it has evolved into a continuous management approach that focusses on the design, implementation, control and optimization of processes. Thereby, a "*Process*" is understood as a defined order of tasks/activities, which are performed by responsible roles to achieve a defined result [7, p. 12].

To describe processes typically two different formats are used depending on the level of granularity of the processes. The so called "*Business Process Model and Notation*" (BPMN) has established as a modeling standard to document *business process flows by so called Business Process Diagrams (BPD)*—rather on a detailed level of granularity focusing on the single tasks incl. Their order, their responsibilities and supporting IT systems [8, p. 99]. In addition the concept of so called *Process Landscapes* has evolved to give an overview on processes with a rather high level of granularity [9, p. 38].

A process landscape is a mean to represent a companies' process hierarchy or process architecture. It typically consists of all processes of a company and provides a high-level overview, i.e. it prescinds from the detailed flow of the processes. Typically multiple processes are assigned to one process group which is then graphically visualized in the process landscape. A process landscape can be decomposed into multiple layers, i.e. more detailed process landscapes. Theoretically, the full process hierarchy can be visualized by using process landscapes for the first levels (typically level 1 to levels 3/4/5) of granularity and business process diagrams for the description of detailed process flows [9, pp. 37 ff.].

A process landscape is not only used to structure the processes per-se and to provide transparency. Instead further benefits can be realized. A process landscape can also be used to identify process optimization and process harmonization potentials. Furthermore, it can even serve as an instrument to support the steering of the company via the steering of the processes [9, pp. 37 ff.].

The realization of the benefits of a process landscape are heavily dependent on the design criteria that were applied to the process landscape. Next to a typical methodical design criteria that groups a company's processes into management, core and support processes, various business-related design criteria can be used to structure the process architecture. Typically, those business-related design criteria are also industry-specific. Figure 1 shows in a nutshell the various business-related design criteria that can be differentiated [9, p. 43].

For each level of the process landscape the design criteria must be determined. Thereby, multiple design criteria can be applied on one level and different design criteria can be used on different levels, e.g. on the first level of the process landscape the "product" dimension is the leading design criteria, whereas on the second level processes are differentiated according to customer segments [9, p. 44 ff.].

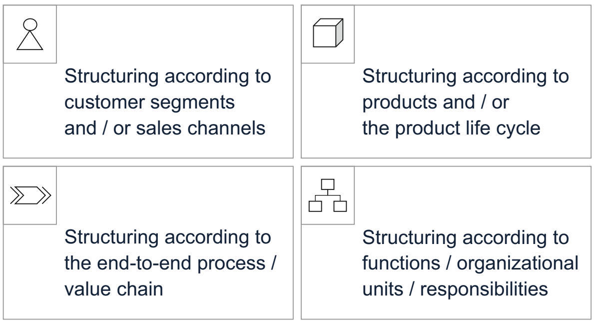


Fig. 1 Business-related design criteria for a process landscape/process architecture

2.3 Interrelation of Capability Management and Business Process Management

Under the assumption that both management approaches—EAM and BPM—are established within a company, it obviously makes sense that relevant content is shared between Capability Management and Business Process Management, so that content, e.g. a process hierarchy, is not maintained twice. Very often, the EA team is responsible for the capability map, whereas the BPM team is responsible for the company wide process landscape.

Usually, that means that BPM shares its process architecture/process landscape(s) with the EA colleagues. Depending on the EA approach those processes might be related to capabilities and/or applications and/or other elements of the enterprise architecture.

On the contrary, EAM might provide the list of applications to BPM, which is then typically used in the context of the process flow documentation. The use of applications from EA in BPM is a separate scenario, which is not looked at in this contribution.

3 Challenges and Observations in the Context of "Process" and "Capability" from a Practitioner's Perspective

Although from a semantical point of view a "process" (How, i.e. in which order is something done) is different than a "capability" (What is/can be done) [10], in the practical context difficulties and challenges occur. One of the main reasons for this might be the different understanding of a process flow (which describes how, i.e. in which order tasks are performed) in comparison to process elements/process groups included in a process landscape, which often do not contain information about their order, but are structured differently (see also Sect. 2.2). From the author's perspective this often leads to the following two main challenges:

1. In some companies the value add of having both, a process landscape and a capability map is questioned (see also [11]).
2. There is sometimes uncertainty how, i.e. on which levels to relate capabilities and processes.

The first challenge obviously occurs, if the capability map and the high-level process landscape are very similar. Such a similarity occurs, if the same design criteria were applied to both—the capability map and the process landscape. If, for example, the capability map was defined with a functional focus and the process landscape was structured according to functions, both maps/landscapes will most probably look the same. This phenomena is even enforced, if the same naming convention is applied. For capabilities two different naming conventions are suggested [5, p. 100]: (1) use nouns, e.g. "Access control" or (2) Use verb + noun, e.g. "Control access". The recommendation [5, p. 100] and from the author's perspective also the typical practical use with regards to the naming of capabilities is the "noun" naming convention to avoid the same naming as for processes. However, on the process side, where typically the recommendation is to use an active name (i.e. verb + noun), this recommendation is sometimes not followed for the process names in the high-level process landscapes. That is why, in a high-level process landscape one might find the process "Access control" instead of "Control access". As a consequence there might be understanding and acceptance problems for either the capability map or the process landscape (or even both), because the value add of having both instruments does not become clear. The situation might even get worse, in case the two instruments are developed by different teams, e.g. the capability map by the EA team and the process landscape by the BPM team and both deal with the same stakeholders/business owners but presenting slightly different maps/landscapes.

The second challenge is related to the relation of capabilities and processes. Due to the fact that capabilities as well as processes can be detailed into different levels, different scenarios to relate those two elements with each other are available:

- Relate more detailed processes (e.g. level 2 or 3) to a level 1 capability
- Relate detailed capabilities (e.g. level 3, 4 or 5) to a process on a higher level (e.g. levels 1, 2 or 3)
- Relate capability and process of the same level with each other, e.g. level 1 capability with a level 1 process

From the author's perspective, questions related to the relation of processes with capabilities and vice versa typically occur, if the underlying use case for the integration of those two elements is not clear.

4 Proposal to Address Those Challenges from a Practitioner's Perspective

4.1 Clear Positioning of a Capability Map and/or a Process Landscape

Taking the first challenge from Sect. 3 into account, it obviously raises the question whether both—a capability map and a process landscape—are needed and helpful within one organization.

In a lot of companies mostly *one instrument*—either be it the *capability map or the process landscape*—is already established before action is taken to develop the *second instrument* as well. If a capability map is already existing as the first instrument, the second instrument would be a process landscape. On the contrary, if a process landscape is already existing as the first instrument, the second instrument would be a capability map.

Already at this early point in time it should be discussed and motivated, whether a second instrument is really needed and helpful or whether the existing (first) instrument can fulfill the raised requirements. The more established the existing instrument is already, the more the second instrument should be questioned.

But what does it mean, that an instrument is already well established? From the author's perspective a capability map or a process landscape is well established, if

- it fulfills the as-is requirements/use cases,
- an ownership concept has been established and implemented,
- the instrument is known and accepted, especially by its stakeholders, but also within the overall company,
- the instrument is rather stable, i.e. multiple update cycles have already been undertaken.

If those aspects are given for the existing instrument, it is rather recommended to stick to the existing one and align on how to integrate the new requirements/use cases.

On the contrary, if either the existing instrument is not well established yet or if it does not seem to be possible to integrate the new requirements/use cases into the existing instrument, it might be worth to consider the establishment of a second instrument. In such a situation it is inevitable to clearly position the second instrument. To realize this, it is proposed to highlight the criteria that distinguish the second instrument from the first instrument and to summarize the overall value add of having a second instrument. Such distinguishing criteria could be:

1. different structuring criteria, e.g. end-to-end processes on the process landscape vs. functional structuring of the capability map,
2. different time perspective, e.g. a process landscape representing the as-is situation of the company vs. a capability map also taking into consideration future capabilities for new business models or to implement strategic goals for which processes and applications still have to be developed,
3. different topics that occur in multiple processes, where e.g. an overview to steer different transformation initiatives is needed. Such "topics" might be represented as capabilities, e.g. workflow management, document management or digitalization as a very high-level topic.
4. different group/company structure, e.g. different process landscapes for different entities of a group and one "abstract" capability map on group level.

Furthermore, it is not only critical to clearly distinguish the two instruments, but also to properly establish the second instrument according to the above mentioned aspects. This is especially relevant with regards to the ownership concept. Also for the second instrument an ownership concept has to be established and it has to be made sure that those owners are not in conflict with the owners of the first instrument.

Coming back to the first observation in Sect. 3, it can be summarized that questioning the value add of having both—a capability map and a process landscape—and/or leading many discussions about the naming of capabilities and processes, reveals that the design of both instruments and their distinguishing criteria have not been thoroughly thought through yet.

4.2 Different Levels of Capabilities and Processes and Their Linking

Capabilities as well as processes can be decomposed into more granular capabilities and processes. As a result a hierarchy of capabilities or processes can be achieved, typically starting with "Level 1" (L1) and depending on the depth being detailed to "Level 2" (L2) and "Level 3" (L3) as well as sometimes even to "Level 4" (L4) or "Level 5" (L5).

Within Business Process Management, process landscapes typically cover levels 1 to 3, sometimes also L4 or L5, before a detailed process flow is described (see also Sect. 2.2). On the contrary, capability maps typically cover L1 and L2 capabilities (see also Sect. 2.1), but might be detailed even further to the levels 3 to 5.

Already per management approach it has to be decided depending on the underlying requirements/use cases, how deep the hierarchy of capabilities or processes should be developed and maintained. If both instruments are used within one organization this has to be reviewed once again taking into account the level of depth of both instruments. This can be illustrated with the help of an example: If a company does not have a capability map but an established process management approach, one would rather expect process landscapes from L1 to L3/4/5, probably with a deep dive into (at least some) detailed process flows. If on the contrary, the company has an established capability map on L1 and L2, it would may be only have detailed process landscapes on L3/4/5 connecting to detailed process flows.

Next to the level of detail, there is sometimes uncertainty about how, i.e. on which levels capabilities should be related to processes (and/or vice versa) if both instruments are used (see question 2 in Sect. 3). Theoretically, four different scenarios can be differentiated (see also Fig. 2):

1. More detailed processes are related to high-level capabilities (e.g. L2/L3 processes to L1/L2 capabilities)
2. More detailed capabilities are related to high-level processes (e.g. L3/4/5 capabilities to L2/3 processes)
3. Processes and capabilities on the same level are related to each other (e.g. processes and capabilities on L1)
4. Processes and capabilities are not related to each other although both instruments exist within one organization

Scenario 1 is typically implemented, if the requirement/use case "How are capabilities implemented?" is relevant for the organization. In such a use case, typically not only processes, but also applications and may be also other elements of the enterprise architecture are related to capabilities. The advantage of such a

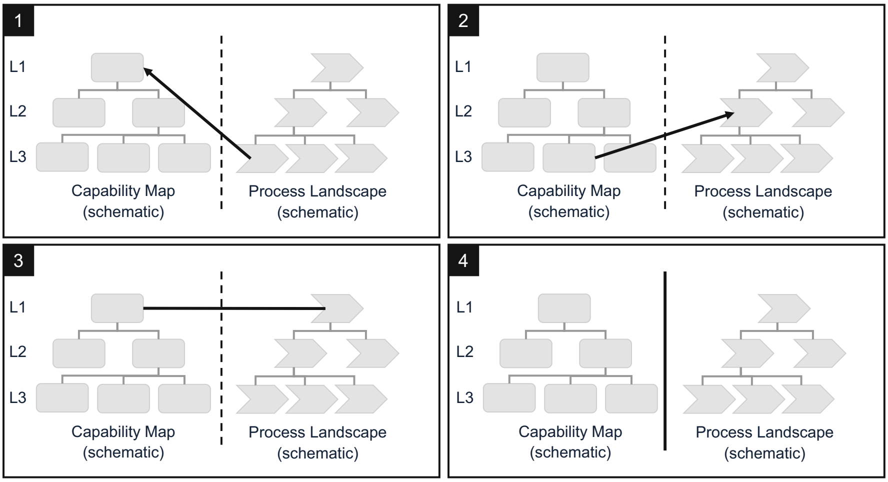


Fig. 2 Different scenarios to link capabilities and processes

scenario is, that it is possible to think about necessary capabilities first, especially in the context of strategic planning, without having the need of immediately thinking about the implementation. However, once implementation is/gets clearer, processes and applications can be linked. Disadvantages will appear, if the distinguishing criteria from Sect. 4.1 and the different levels of detail have not properly been considered. As an example an insurance company is taken, that implemented a (functional-oriented) capability map on L1 and L2 as well as functional-driven process landscapes from L1 to L3. If for example, a L2 claims process (e.g. "Set-up new loss") belongs to the L1 process "Claims Management", but is at the same time linked to the capability "Claims Management", there is no value-add of such an exercise.

Scenario 2 is typically implemented, if the requirement/use case "Which building blocks/capabilities are implemented by a process?" is relevant for the organization. The advantage of such a use case can be seen in a better support for the identification of adequate IT support/implementation. As a disadvantage it can be mentioned, that the maintenance effort for such detailed capabilities and their relation to processes might be very high in large/group organizations. Typically, this goes very closely together with a rather very high complexity of such a scenario.

From the author's experience, scenario 3 is also sometimes discussed. However, a clear use case behind it, is not seen or known. There might be a situation, where L1 end-to-end processes are linked to L1 capabilities. However, the value-add is not recognizable. Hence, the recommendation is not to link processes and capabilities on the same level.

Scenario 4 is an option, if the decision was made to establish both instruments on the one side, but on the other side the value-add of scenarios 1 and 2 is not perceived as high enough. This might be the case, if e.g. a capability map (on L1 or max L2) is used as a strategic instrument to plan the strategic transformation towards the companies' objectives (e.g. projects are related to capabilities to make the transformation with regards to the capability implementation/improvement transparent). In addition, process landscapes are used in parallel to provide an oversight on the overall process architecture and to serve as an entry point into the process world. In such a scenario there still might be an interrelation between Business Process Management and Enterprise Architecture Management, e.g. via the process—application relation. If on the EA side, applications would also be related to capabilities, consequently there would be an indirect relation between processes and capabilities via the application element.

5 Summary and Outlook

The contribution at hand focusses on one possible interface between Enterprise Architecture Management and Business Process Management—both being established management approaches within large companies across all industries. The interface relates to the elements "capability" and "process", whereas capabilities are typically managed as one part of the enterprise architecture and processes typically being managed as part of process management. Although the establishment and tradition of both approaches goes back two to three decades, there are still open questions when it comes to the practical appliance and usage of those elements/concepts within organizations.

Two open questions are covered by this contribution: (1) What is the value add of having a capability map provided by EAM on the one hand and a process landscape provided by BPM on the other hand? (2) On which levels of granularity can processes be linked to capabilities and/or vice versa?

To address those questions, Sect. 2 sets the frame by explaining the relevant fundamentals of Enterprise Architecture Management incl. Capability Management as well as Business Process Management.

Based on this, Sect. 3 details the questions above and explains from a practitioner's perspective the background and challenges that occur in practical usage. For capability maps and process landscapes it is identified, that challenges with regards to practical acceptance especially occur, if for both maps same design criteria are used. This is even enforced, if the same naming pattern is applied. When it comes to the relation of capabilities and processes different scenarios are outlined. Challenges and discussions within a company with regards to the relation typically occur if the underlying use case for the interrelation of capabilities and processes is not clear.

Section 4 provides a two-folded proposal from a practitioner's point of view on how to address those challenges: (1) First of all a clear positioning of a capability map and/or a process landscape is needed. If one instrument (either a capability map or a process landscape) is already established, the recommendation is to rather stick to this established instrument. The paper presents some criteria to identify, whether an instrument is already well established. If one instrument is not sufficiently established yet and/or the additional requirements/use cases cannot be fulfilled by the existing instrument, the second instrument can be established. However, in such a situation a clear positioning of both instruments is needed and therefore the paper at hand proposes four distinguishing criteria. From practical experience, it seems to be promising if for example different structuring criteria are used, e.g. a functional-oriented capability map and an "end-to-end process"-driven process landscape. (2) Furthermore, it is necessary to reflect on the different levels of capabilities and processes and how they can be related to each other. To address this topic four scenarios including their underlying use cases are presented.

Although this proposal supports a structured discussion about the value add of having a capability map and a process landscape and their relation to each other, further research and contributions are needed. Further case studies about established approaches would be helpful—either where both instruments are in place including information about their distinguishing criteria or where an explicit decision for one instrument was made including information about the implementation of the new requirements that called for the second instrument. Finally another area for further research can be seen in the "process" understanding of ArchiMate® with its different elements (see Sect. 2.1) and the alignment of this understanding with Business Process Management.

References

1. Karagiannis, D., Moser, C., Helmes, A.: Geschäftstransformation—Eine Notwendigkeit. In: Karagiannis, D., Moser, C., Helmes, A. (eds.) Benutzerzentrierte Unternehmensarchitekturen, pp. 13–31. Springer Vieweg, Wiesbaden (2020)
2. Hilliard, R.: IEEE-Std-1471-2000. Recommended Practice for Architectural Description of Software-Intensive Systems. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e74f0b588e95d5e3c5c5176617026f48a63655d5 (2000). Accessed 23 Dec 2023
3. Helmes, A., Hauer, D.: Das Zusammenspiel zwischen TOGAF, ArchiMate und EA-Szenarien. In: Karagiannis, D., Moser, C., Helmes, A. (eds.) Benutzerzentrierte Unternehmensarchitekturen, pp. 33–58. Springer Vieweg, Wiesbaden (2020)
4. Aldea, A., Iacob, M.E., Lankhorst, M., Quartel, D., Wimsatt, B.: Capability-based planning: the link between strategy and enterprise architecture, white paper. The Open Group. https://publications.opengroup.org/white-papers/archimate/w16c (2016). Accessed 14 Jan 2024
5. Koç, H.: Capability-portfolio-management. In: Karagiannis, D., Moser, C., Helmes, A. (eds.) Benutzerzentrierte Unternehmensarchitekturen, pp. 91–117. Springer Vieweg, Wiesbaden (2020)
6. The Open Group 2023: Business capability planning. https://pubs.opengroup.org/togaf-standard/business-architecture/business-capability-planning.html (2023). Accessed 23 Dec 2023
7. Gericke, A., Bayer, F., Kühn, H., Rausch, T., Strobl, R.: Der Lebenszyklus des Prozessmanagements. In: Bayer, F., Kühn, H. (eds.) Prozessmanagement für Experten, pp. 11–33. Springer Gabler, Berlin, Heidelberg (2013)
8. Murzek, M., Rausch, T., Kühn, H.: BPMN als Bestandteil der BPMS Modellierungsmethode. In: Bayer, F., Kühn, H. (eds.) Prozessmanagement für Experten, pp. 93–113. Springer Gabler, Berlin (2013)

9. Bayer, F., Appelhans, L., Wolf, E.: Prinzipien für die Gestaltung der Prozessarchitektur. In: Bayer, F., Kühn, H. (eds.) Prozessmanagement für Experten, pp. 37–55. Springer Gabler, Berlin (2013)
10. Spacey, J.: Capability vs process. https://simplicable.com/IT/capability-vs-process (2023). Accessed 20 Jan 2024
11. Campbell, A.: Capability maps vs process maps: horses for courses. https://ashridgeonoperatingmodels.com/2016/04/14/capability-maps-vs-process-maps-horses-for-courses/ (2016). Accessed 20 Jan 2024

Ontology-Based Metamodelling, Modelling and Application Development

Knut Hinkelmann

Abstract Using three dimensions of the knowledge space for conceptual modeling, graphical models and logic-based ontologies are compared. Their integration results in ontology-based modelling and metamodeling. Models can be easily created and understood by humans. At the same time, the representation as an ontology allows for machine interpretation. It is shown how ontology-based metamodelling can overcome disadvantages of the Meta Object Facility and the Model-driven Architecture for application development, knowledge-based systems, model validation and knowledge management.

Keywords Conceptual modeling · Ontology · Metamodeling · Model-driven Engineering

1 Introduction

Conceptual modelling is the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication [1]. Ontological theories provide the basic constructs for representing real-world phenomena in conceptual models [2]. Guizzardi [3] elaborated on characterizations of ontology, conceptualization and metamodel, as well as on the relations between them and discussed criteria for a suitable ontology representation language.

Before creating a model, the modeler has a mental model of what should be in the model. In Fig. 1 this is called *Abstraction*. It is "articulated" using domain concepts, which comprise the *domain conceptualization*. Conceptualizations and abstractions are immaterial; they only exist in the mind of the modeler. To be

K. Hinkelmann (✉)
FHNW University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland

OMiLAB gGMBH, Berlin, Germany
e-mail: knut.hinkelmann@fhnw.ch

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_5

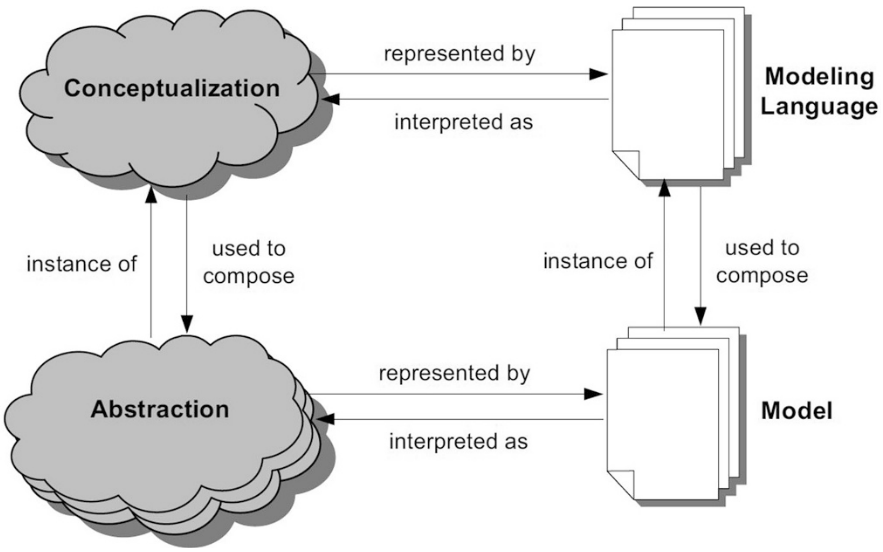


Fig. 1 Relations between Conceptualization, Abstraction, Modeling Language and Model [3]

communicated, analyzed and used, they must be captured, i.e. represented in terms of some concrete artifacts. The artifact of the abstraction is called the model, the artefact of the conceptualization is the modelling language.

Not the complete conceptualization needs to be represented in the modelling language and not all of the abstraction needs to be represented in the model. We represent in the model only that part of reality that is needed for a specific purpose and leave out things that are either *not relevant* or that we assume are *shared by all users* of the model.

For a successful use of the models, we assume that modelers and model users share the *same conceptualization*. This is particularly important for that part of the conceptualization and abstraction that is not represented in the model.

The conceptualization of the domain can be represented in different modelling languages, which then results in different models. In this paper, I deal with two kinds of modelling languages for conceptual modelling—*graphical (diagrammatic) models* and *logic-based ontology*—and how the choice of the language influences the usage of the models.

The structure of the article is based on the research contributions of Dimitris Karagiannis. Graphical and ontology-based modelling are analysed using three of the four dimensions of the knowledge space of Karagiannis and Woitsch [4]. First, the levels of meta-modelling are presented as a further development of the Meta-Object Facility (MOF) and applied to ontology-based and graphical modelling. These are then compared in terms of their potential for interpretation by humans and machines and an integration is proposed. Finally, the potentials of ontology-based modelling are demonstrated to overcome some weaknesses of the model driven architecture and how it can be used for further applications.

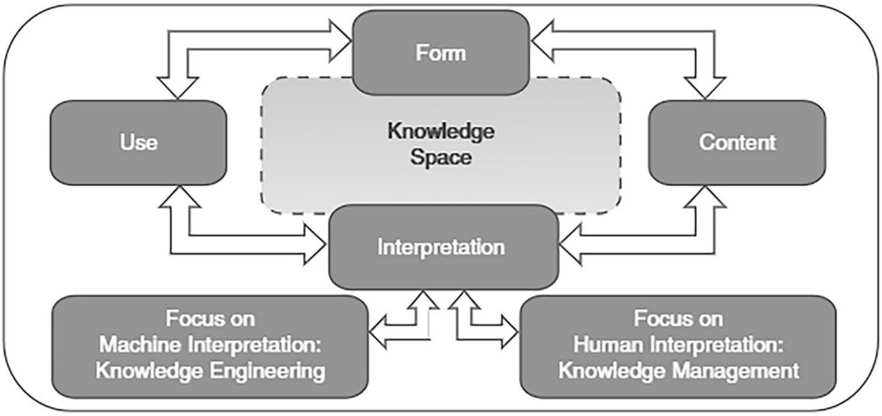


Fig. 2 Dimensions of the knowledge space [4]

2 Knowledge in Models

I use the knowledge space of Karagiannis and Woitsch [4] to explain the commonalities of the two conceptual modelling approaches (see Fig. 2). The knowledge space was originally described for business process management but can be applied for other usages of conceptual modelling, too.

The knowledge space consists of four dimensions:

- The *content* is seen as the domain in which the models are applied. The content of the model covers that part of the (envisioned) reality, that is required for its use.
- The *form* represents the syntax and semantics of the models. Here is the main distinction between graphical and ontological models. Symbols in graphical models are pictorial signs. Ontological models consist of expressions, typically using a subset of first-order logic, where symbols are sequences of characters.
- The *interpretation* is closely related to the form. While graphical models are well-suited for human interpretation, ontological models enable automated reasoning and thus can be interpreted by machines.
- The *use* determines the application of the model and the knowledge such as documentation, transformation, analysis, simulation, decision making, or application development.

3 Modelling Languages for Conceptual Modelling

Karagiannis and Kühn [5] consider a modelling language as a component of a modelling method (see Fig. 3). The other components are the modelling procedure and the mechanisms and algorithms, that are used during modelling and that can be

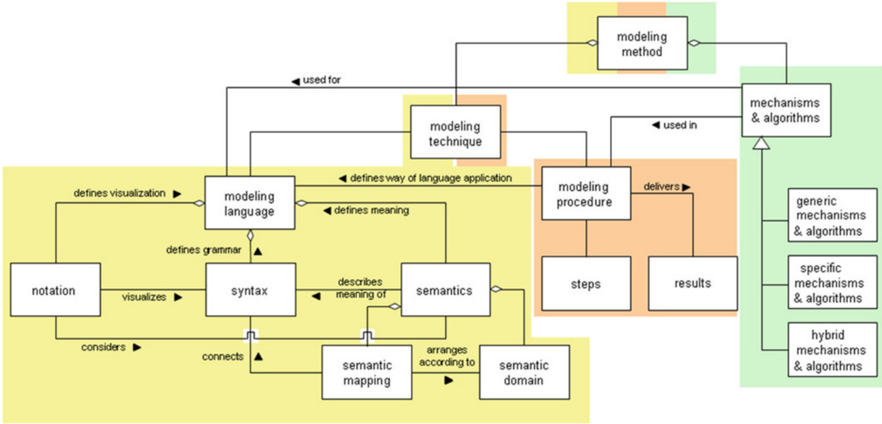


Fig. 3 Components of modelling methods [5]

used for analysing, transforming, or executing the models. Therefore, the decision for the choice of a modelling language must not only consider the expressiveness but also the fit for the intended usage.

A modelling language consists of semantics, syntax and notation. The conceptualization (see Fig. 1) determines the semantics of the modelling language. The syntax is determined by the symbols of the modelling languages as well as rules determining, what are valid combinations of the symbols [3]. In *sentential modelling languages*, the symbols are sequences of characters and the abstract syntax is typically defined as regular expressions. Logic-based ontology belong to the category of sentential modelling languages. In *graphical languages* the symbols are pictorial signs and the abstract syntax is defined in the metamodel. The concrete syntax (called notation in Fig. 3) is the concrete appearance. For graphical languages this are the shapes of the symbols, for sentential languages it is the format. For example, ontologies represented in RDFS [6] can be shown as Turtle or XML notation.

3.1 Graphical Modelling Languages

The Meta-Object Facility (MOF™) [7] is the foundation of OMG's industry-standard environment where models can be exported and imported by applications, stored in a repository, transformed, and used to generate application code.

MOF consists of a hierarchy of four model levels (see Fig. 4). The bottom level, M0, holds the objects of the reality. The next level, M1, holds the conceptual user model representing the objects of level M0. Level M2 is a model of the information at M1 and is referred to as a metamodel. It specifies the modelling elements that can be used for modeling on level M1. In Fig. 4 UML Class, Instance and Attribute

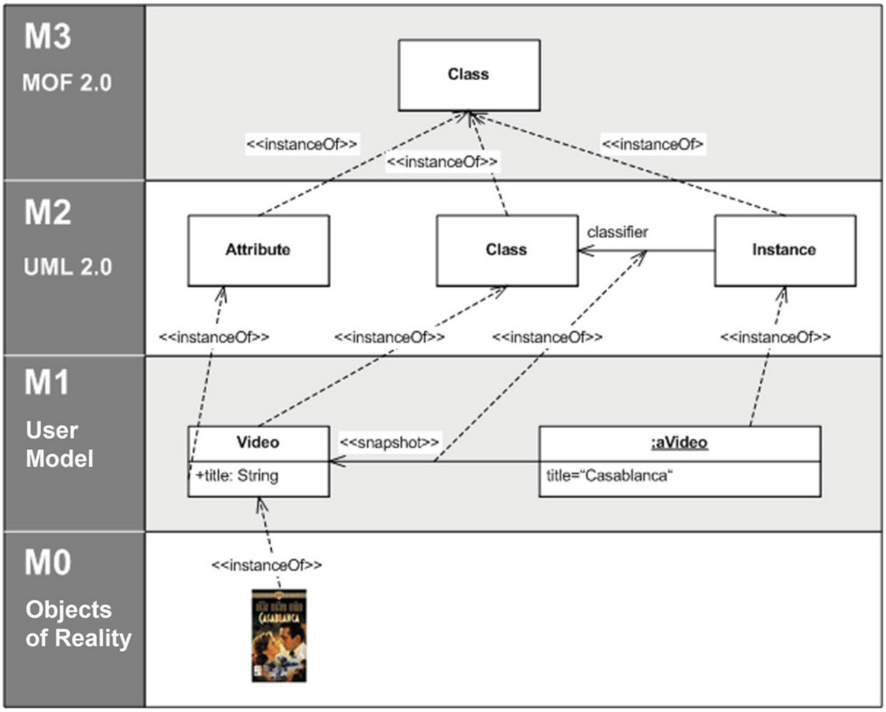


Fig. 4 Meta-object facility (Original graphic from https://commons.wikimedia.org/wiki/File:M0-m3.png)

are used to represent that a video with title "Casablanca" is an instance of class Video. The level M3 holds a model of the information at M2, and therefore is the meta-metamodel, often also referred to as the Meta-Object Facility.

As shown in Fig. 4, level M1 contains both, the conceptualization of a domain, represented as classes, and their instances. This approach is criticized by Atkinson and Kühne [8], who introduce the distinction between linguistic and ontological instance-of relationships.

The use of general purpose modelling languages like UML has been identified as another disadvantage of model-driven engineering based on MOF [9]. This can be overcome by using domain-specific modelling languages, where the domain conceptualization is defined on level 2 and the models are on level 1. This is the reflected in the metamodel layers defined by Karagiannis and Kühn [5] (see Fig. 5).

Here, in contrast to MOF, the second level contains the conceptualization of a domain-specific modelling language (the classes for objects and relations) and the models on level 1 consist of instances of these classes (see Fig. 6)—corresponding to ontological instance-of relationship in the sense of Atkinson and Kühne [8].

The Agile Modelling Method Engineering (AMME) is an approach for the development and adaption of modelling methods [11]. It is based on the metamodel

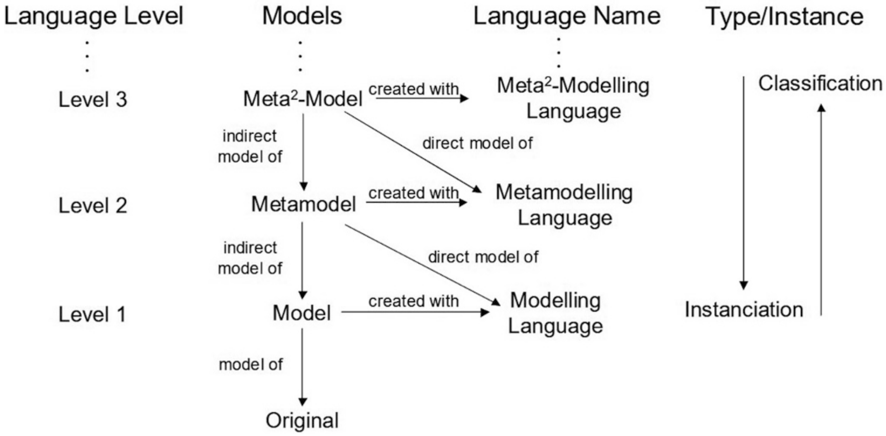


Fig. 5 Metamodel layers [5], based on [10]

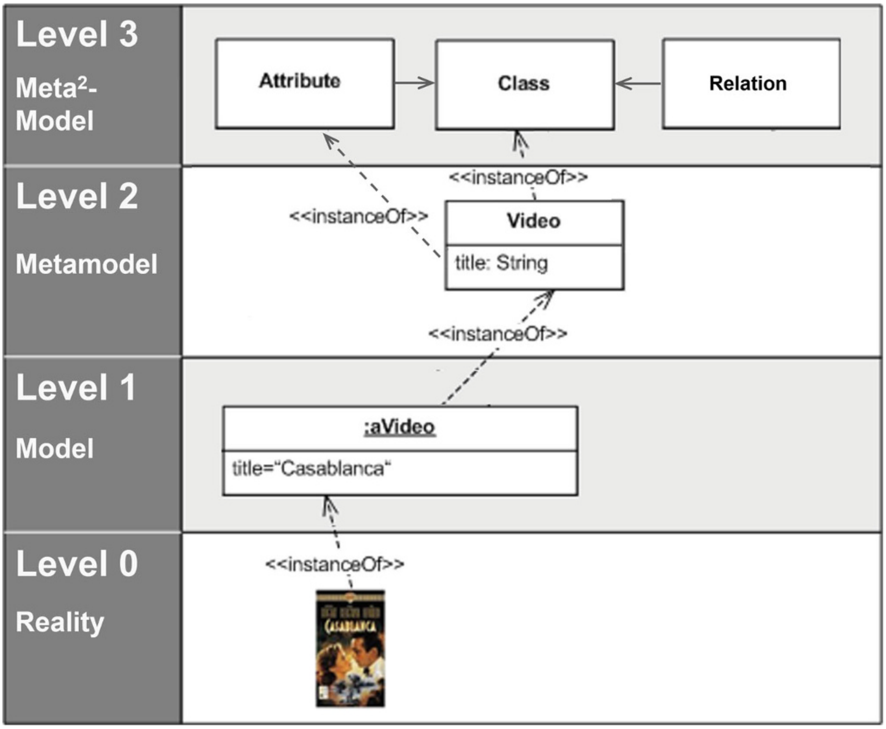


Fig. 6 Metamodel layers with example

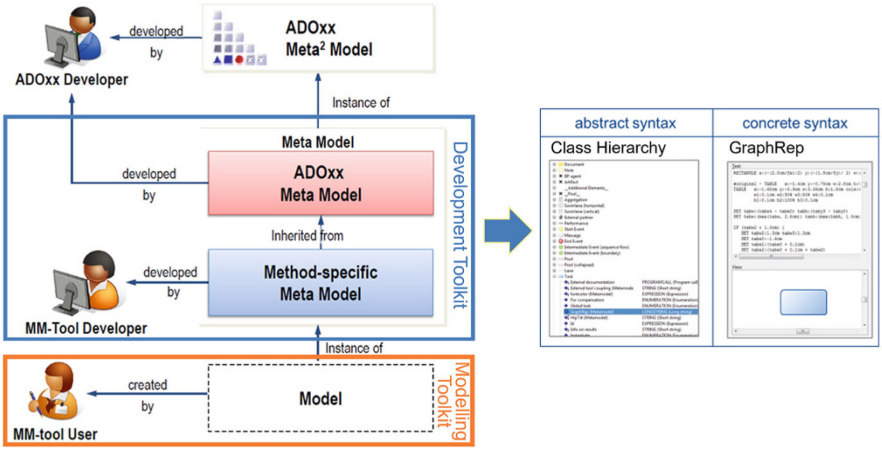


Fig. 7 Metamodelling in ADOxx

layers of Fig. 5 with predefined meta-metamodel and specific toolkits for modelling and metamodeling. The AMME framework is instantiated by the Open Models Initiative Laboratory OMiLAB[1] as support for research projects and communities and realized by the ADOxx metamodeling platform.[2] Figure 7 shows on the left the architecture of ADOxx. It consists separate toolkits for metamodeling and modelling. On the right is an excerpt of the metamodel for BPMN in ADOxx showing the class hierarchy representing the abstract syntax and, on the right, the concrete syntax for the class Task.

The book serious about domain-specific conceptual models edited by Karagiannis et al. collects a large number of domain-specific conceptual modelling methods based on this approach [12, 13].

3.2 Logic-Based Ontologies

The word "ontology" has different meanings in different communities [3, 14]. In Artificial Intelligence, the term ontology is used for a kind or knowledge representation which originated from semantic networks, a graphical representation consisting of nodes representing objects or category names and links connecting them.

In his seminal paper about the epistemological status of semantic networks [15] Brachman introduced an additional epistemological layer allowing to distinguish

[1] https://www.omilab.org/

[2] https://adoxx.org/

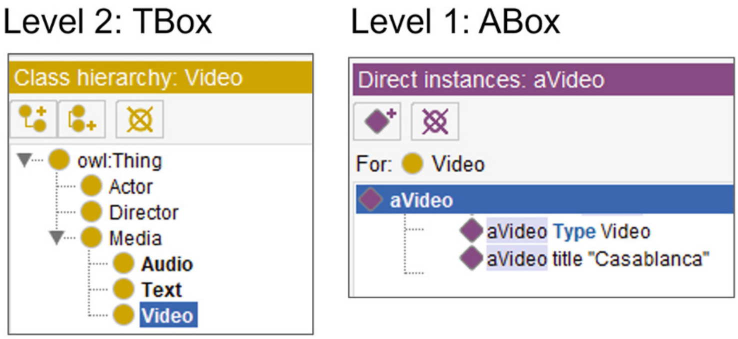


Fig. 8 Sample ontology for the video example (screenshots from Protege (https://protege.stanford.edu/))

between types of concepts and relations. This led to the development of description logics [16], which make a distinction between the terminological and assertional representation and reasoning—often referred to as TBox and ABox. Individuals represented in the ABox are instances of concepts represented in the TBox (also called the schema of the ontology). The first knowledge representation language based on description logic was KL-ONE [17]. In the meantime, the ontology-language of the semantic web OWL [18] is a kind of quasi-standard for description logics.

The TBox/ABox distinction is in line with the separation of levels 1 and 2 in metamodeling (see Fig. 5): The TBox corresponds to the metamodel on Level 2 and the ABox corresponds to the model on Level 1. Figure 8 shows the video example represented as an ontology.

The distinction between ABox and TBox is primarily useful for implementing efficient and decidable reasoners. RDFS [6] is an ontology language that allows for various levels of ontological instantiation and thus satisfies the requirements of [8].

4 Interpretation of Models

Model interpretation requires knowledge. One can distinguish between two kinds of knowledge that is needed for model interpretation:

- knowledge about the conceptualization, i.e. the syntax and semantics of the predefined concepts and relations.
- knowledge about specific applications, i.e. the meaning of the instantiations.

Fig. 9 Interpretation of model elements

4.1 Interpretation of Graphical Models

For graphical models, the knowledge for the correct interpretation is in the mind of the human. The conceptualization is specified in the modeling language and both the human modeler and user are expected to know the syntax and semantics of the concepts. When interpreting a model, the human user recognizes the notation of the concepts and can interpret its meaning.

Figure 9 shows two graphical model elements. On the left is an element of the ArchiMate language for enterprise architecture. From the shape, the human user can see that the element represents an application component. The second element in Fig. 9 can be interpreted as a manual task in BPMN, which again can be recognized from its shape. The concepts and shapes of application component and subprocess are defined in the metamodels of ArchiMate and BPMN, respectively.

Knowing the metamodel concepts, however, is not sufficient for an interpretation of the model. Additional knowledge about the application domain is required to interpret the designations of the model elements. In the examples of Fig. 9, it is expected that the user knows that "Evento" is the student administration system used ad FHWN and that "Fettuccine" is an Italian pasta.

Summarizing, it can be said that graphical models are adequate for human interpretation, but requires some knowledge management effort: Users must be trained so that the know the meaning of the modelling elements. Modelers are responsible for choosing the designations such that they are unambiguous and are understood in the same way by all users of the model. Ambiguous designations and missing knowledge are sources of misinterpretation and misunderstanding.

4.2 Machine Interpretation of Ontological Conceptualization

Logic-based ontological modelling languages allow for modelling not only the modelling elements but also additional knowledge about the application domain. For the example of Fig. 9, the ontology would not only contain classes for applications component and manual task but also contain the knowledge that Evento is an information systems and that Fettucine is a kind of meal.

A more realistic scenario is shown in Fig. 10. It shows the ontological model of a task "send invoice". On the left is a subset of the class hierarchy representing the conceptualization of business processes,. On the right is an ontological representation of the APQC Process Classification Framework, which represents knowledge about the application domain.. "Send invoice" is an instance of the class "UserTask"

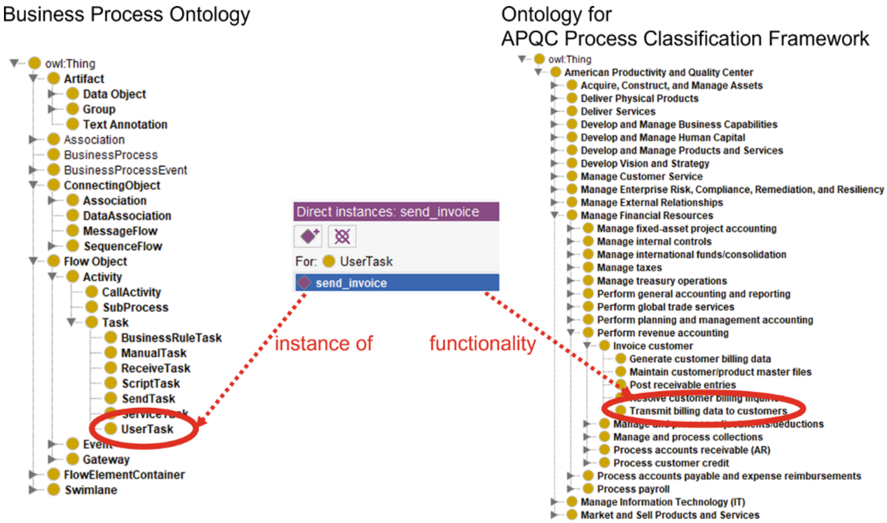


Fig. 10 Ontological conceptualization for business process modelling

and has a reference to "Transmit Billing Data", which represents the functionality of the task.

While ontological models can be interpreted by machine, they are not adequate for use by humans. Although machine interpretation of models can be of advantage for some explications, representing the additional application knowledge requires additional effort. It particularly makes sense, if one can rely on already existing knowledge representation, as in the example above, where the predefined APQC process classifications was reused.

4.3 *Combining Human and Machine Interpretation*

As shown in the previous sections, both human and machine interpretation of models have advantages and disadvantages. This is why a combination of graphical models and ontologies can make sense. Semantic Lifting as developed by Karagiannis and Höfferer [19] extends metamodel with application knowledge (see right side of Fig. 11).

Semantic lifting has the disadvantage that models and ontologies are represented in different forms and in different environments. Ontology-based metamodeling allows to overcome this disadvantage by representing also the metamodel as an ontology (Fig. 12).

Ontology-based metamodelling has been implemented in the AOAME tool [20], which provides a single environment for metamodelling, modelling and ontology development. Modelling elements are directly created as instances of the

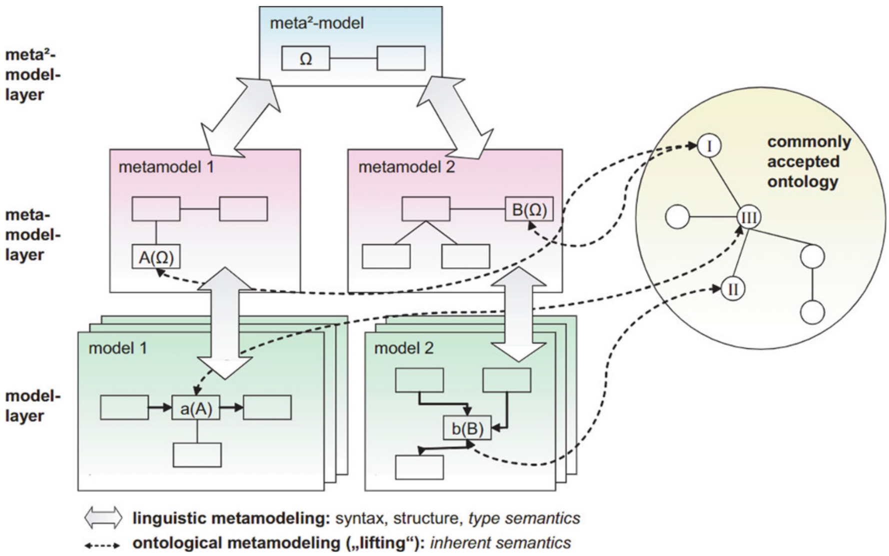


Fig. 11 Semantic Lifting: Extending metamodels with an application domain ontology [19]

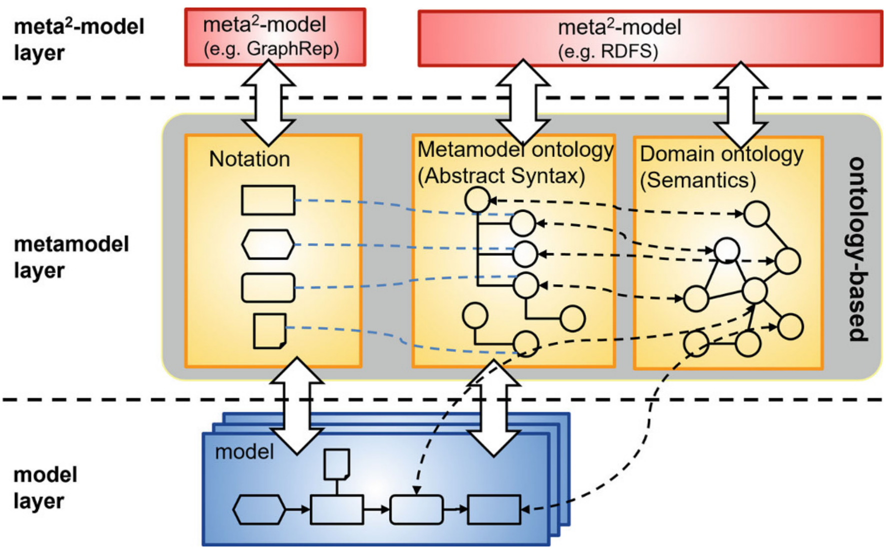


Fig. 12 Ontology-based metamodeling

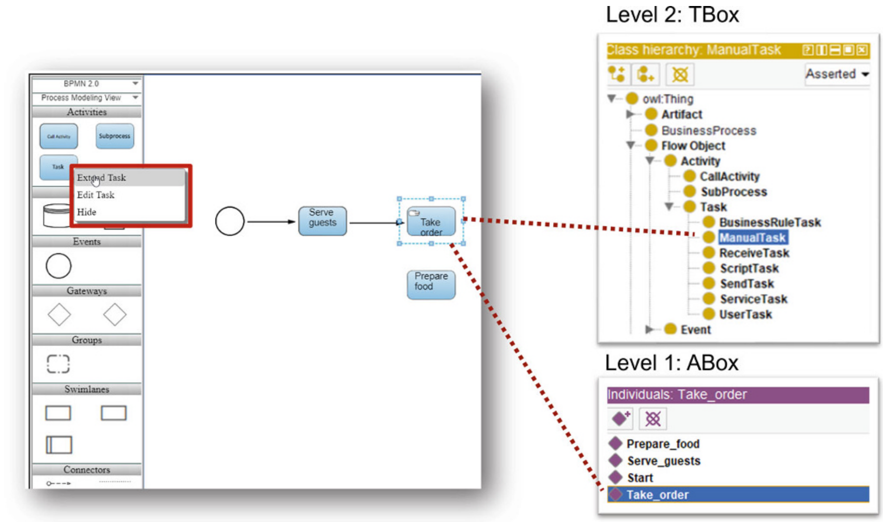


Fig. 13 AOAME modelling environment and ontology

classes of the modelling language (Fig. 13). It additionally extends the agility of metamodelling as it allows to expand the metamodel ontology on the fly during modelling (see framed element in Fig. 13).

5 Use of Ontology-Based Metamodelling

Model Driven Architecture (MDA)[3] is a standard from OMG to support model-driven engineering . MDA provides guidelines for structuring software specifications that are expressed as models. MDA is related to multiple standards, in particular UML and MOF. This has resulted in drawbacks, which are already described above. Metamodelling can contribute to overcoming these drawbacks. By using domain-specific metamodels, rules for model transformation can be specialized and reused. In the following I discuss potential advantages of ontology-based modelling for model-driven engineering.

[3] https://www.omg.org/mda/

5.1 Model-Driven Software Development

One idea of MDA is that code can be generated by transforming model down the different levels of abstraction. MDA distinguishes between models on three level of abstraction: Computation-independent Model (CIM), the Platform-independent model (PIM), and the Platform-specific model (PSM).

Figure 14 shows an adaptation of model-driven architecture, which uses the layers of metamodeling instead of the MOF layers. The conceptualization is on level 1, with metamodels for the of the models CIM, PIM, and PSM. I used dotted lines between the metamodels to indicate that they are not strictly separated because all models deal with the same domain—just from different perspectives and different levels of abstraction.

- The CIM provides an understanding of the system's purpose and requirements. It captures the business goals, processes, and rules, serving as a bridge between the business domain experts and the software developers.
- The PIMs focus on the logical structure and behaviour of the system. They allow for the exploration of various design options, enabling system architects to make informed decisions without being tied to a specific platform.
- The PSMs take into account the specific technologies that will be used for implementation. This level of abstraction allows for the generation of code, configuration files, and other artifacts that are specific to the chosen platform.

Together the models on these abstraction layers facilitate the development of robust and adaptable software systems.

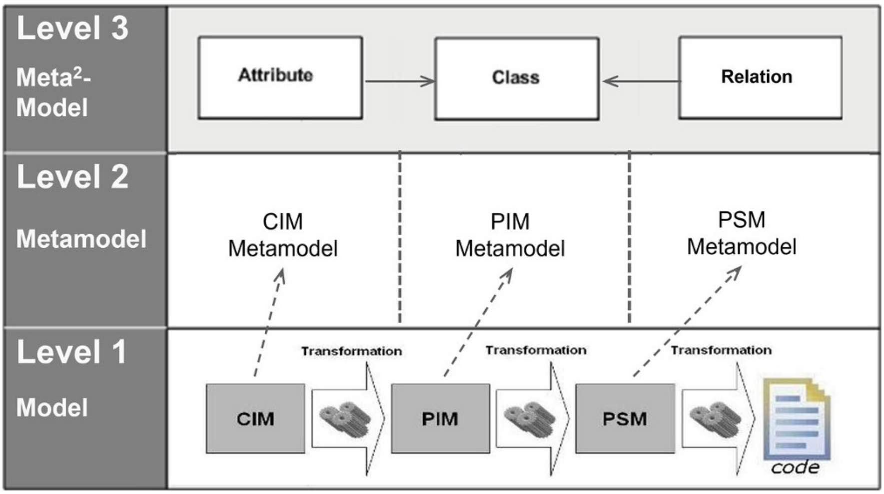


Fig. 14 Adapting Model-Driven Architecture with domain-specific modelling

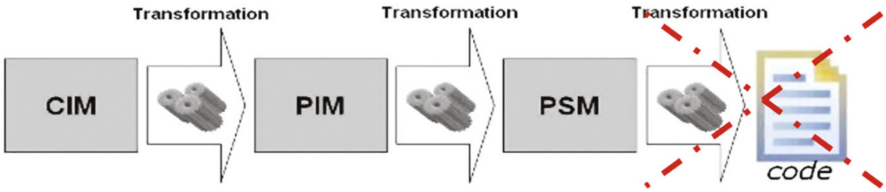


Fig. 15 Model transformation for the development of knowledge-based systems

Transforming models from the higher-level, computation-independent models to code is at the core of MDA. Automating this transformation is a challenge. To support model transformation, the OMG has standardized Foundation UML, which is influenced by Executable UML [21]. Ontology-based modeling and metamodeling offers advantages for model transformation.

- While fUML is a general-purpose language, using ontology-based modelling and meta-modelling allow for creating domain-specific modelling languages.
- Adding ontologies about the application domain can allow for defining domain-specific transformation rules, which can be reused for multiple projects. As an example, consider transformation rules for digitalisation of business processes, which can be standardized based on the standard BPMN metamodels for business processes.

A special case is the development of knowledge-based systems. As the models can be interpreted by machines, the platform-specific model is executable itself. This means that the code-generation step is not needed, because the model itself can be used for automated reasoning (see Fig. 15).

5.2 *Model Validation*

As ontology-based modelling languages can be interpreted by human and machine, the models and their transformations can be automated and are at the same time human-understandable, which allow for automated validation of the models. In [22, 23] we describe an approach for validation of enterprise architecture models. Enterprise architecture principles can be represented on various abstraction layers corresponding to the distinction between CIM and PIM. On the CIM level, the architecture principles are represented using structured English notation as specified in SBVR [24]. The enterprise models are represented using the extended ArchiMEO ontology [25] and annotated with domain knowledge and formal representation of derivation rules between architecture views. This corresponds to the PIM level. Thus, the validation combines ontology-based enterprise architecture models with domain knowledge and allows for automated validation of architecture models for conformance with enterprise architecture principles.

5.3 *Knowledge Management*

While knowledge-based systems automate decision making, there is knowledge that cannot be represented explicitly. This is particularly the case for tacit knowledge which remains in the mind of human experts, because it cannot be made explicit. As already shown in [26], knowledge sources can be described with ontologies, while the PROMOTE approach [27, 28] supports knowledge creation and usage with graphical models. Ontology-based modelling allows for the seamless combination of these approaches by enabling the annotation of the models with knowledge about information, enterprise and the application domain.

6 Conclusion

This paper shows significant contributions of Dimitris Karagiannis to the area of modelling and metamodelling. He laid the foundation for further research. The ontology-based modelling and metamodelling described here is one of these. It extends the findings on metamodelling and semantic lifting. Models can be easily created and understood by humans. At the same time, the representation as an ontology allows further machine evaluation for decision support, model transformation and model-based application development.

References

1. Mylopoulos, J.: Conceptual modelling and Telos. In: Conceptual Modelling, Databases, and CASE: An Integrated View of Information System Development, pp. 49–68. Wiley (1992)
2. Weber, R.: Conceptual modelling and ontology: possibilities and pitfalls. J. Database Manag. **14**(3), 1–20 (2003)
3. Guizzardi, G.: On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. In: Vasilecas, O., et al. (eds.) Databases and Information Systems IV, Papers of the Sevenths International Baltic Conference DB&IS'2006. IOS Press (2007)
4. Karagiannis, D., Woitsch, R.: Knowledge engineering in business process management. In: Michael, V.B.J.A. (ed.) Handbook on Business Process Management 2, pp. 463–485. Springer, Berlin (2010)
5. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: Bauknecht, K., et al. (eds.) Proceedings of the Third International Conference EC-Web at DEXA 2002. Springer, Berlin (2002)
6. W3C: RDF Schema 1.1. World-Wide Web Consortium (2014)
7. OMG: OMG Meta Object Facility (MOF) Core Specification, Version 2.5.1. Object Management Group (2019)
8. Atkinson, C., Kuhne, T.: Model-driven development: a metamodeling foundation. IEEE Softw. **20**(5), 36–41 (Sep–Oct 2003)
9. Den Haan, J.: 8 reasons why model-driven approaches (will) fail. https://www.infoq.com/articles/8-reasons-why-MDE-fails/ Accessed 01 Jan 2024

10. Strahringer, S.: Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In: Pohl, K. et al. (eds.) Modellierung ’98, Proceedings des GI-Workshops in Münster, 11.-13. März 1998. CEUR Workshop Proceeding, vol. 9. https://ceur-ws.org/Vol-9/ (1998)
11. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics, pp. 5–10. Springer (2015)
12. Karagiannis, D., et al.: Domain-Specific Conceptual Modeling. Springer (2016)
13. Karagiannis, D., et al. (eds.): Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools. Springer, Cham (2022)
14. Guarino, N., et al.: What is an ontology? In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, 2nd edn, pp. 1–17. Springer, Berlin (2009)
15. Brachman, R.J.: On the epistemological status of semantic networks. In: Findler, N.V. (ed.) Associative Networks: Representation and Use of Knowledge by Computers, pp. 3–50. Academic, New York (1979)
16. Baader, F., et al.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
17. Brachman, R.J., Schmolze, J.G.: An overview of the KL-ONE knowledge representation system. Cogn. Sci. **9**, 171–216 (1985)
18. W3C: OWL 2 Web Ontology Language Primer, 2nd edn. World Wide Web Consortium (2012)
19. Karagiannis, D., Höfferer, P.: Metamodels in action: an overview. In: Filipe, J., et al. (eds.) ICSOFT 2006—First International Conference on Software and Data Technologies. Springer, Setubal (2006)
20. Laurenzi, E., et al.: An agile and ontology-aided modeling environment. In: 11th IFIP WG 8.1 Conference on the Practice of Enterprise Modelling, pp. 221–237. Springer (2018)
21. OMG: Semantics of a Foundational Subset for Executable UML Models (fUML) Version 1.5. Object Management Group (2021)
22. Montecchiari, D., Hinkelmann, K.: Towards ontology-based validation of EA principles. In: The Practice of Enterprise Modeling, pp. 66–81. Springer (2022)
23. Montecchiari, D., Hinkelmann, K.: Validating enterprise architecture principles using derivation rules and domain knowledge. In: Perspectives in Business Informatics Research, pp. 244–259. Springer Nature Switzerland (2023)
24. OMG: Semantics of Business Vocabulary and Business Rules (SBVR) Version 1.5. Objekt Management Group (2019)
25. Hinkelmann, K., et al.: ArchiMEO: a standardized enterprise ontology based on the ArchiMate conceptual model. In: Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development. SCITEPRESS - Science and Technology Publications (2020). https://doi.org/10.5220/0009000204170424
26. Abecker, A., et al.: Toward a technology tor organizational memories. IEEE Intell. Syst. Appl. **13**, 3 (1998). https://doi.org/10.1109/5254.683209
27. Hinkelmann, K., et al.: PROMOTE—Methodologie und Werkzeug für geschäftsprozessorientiertes Wissensmanagement. In: Abecker, A., et al. (eds.) Geschäftsprozessorientiertes Wissensmanagement, pp. 65–90. Springer, Berlin (2002)
28. Woitsch, R., Karagiannis, D.: Process-oriented knowledge management systems based on KM-services: the PROMOTE approach. Int. J. Intell. Syst. Account. Finance Manag. **11**(4), 253–267 (2002)

Pushing the Boundaries of Process-Oriented Quality Management Through Conceptual Modeling

Florian Johannsen

Abstract Quality management must continuously adapt to new business trends and technical progress. However, there is often a lack of time and resources to deal with the development of new methods for quality management. For this reason, the focus in practice is primarily on the question of how established quality management methods can be specifically adapted and expanded for new fields of application. At this point, conceptual modeling and metamodeling are valuable means to integrate new concepts with established quality techniques and methods to further develop the discipline. This paper highlights the topics of "smart services" and "environmental sustainability" and shows how modeling-based solutions for these fields can be purposefully created with the help of metamodeling.

Keywords Quality management · Smart services · Environmental sustainability

1 Introduction

These days, enterprises face numerous challenges such as constantly rising customer expectations, high energy costs, skills shortages, fragile supply chains or the need to integrate new technologies (e.g., artificial intelligence, immersive media, etc.) just to mention a few [1, 2]. Considering this, the redesign of a company's business processes to keep pace with the changing market conditions is of utmost importance [3, 4]. For this purpose, the value of process-oriented quality management (PQM) (cf. [5]) is recognized in both practice and literature as a means to support digital transformation initiatives (e.g., [6]), improve process performance (e.g., [7, 8]) or reduce greenhouse gas emissions and energy consumption (e.g., [9]) among others. Various methods such as Six Sigma, Lean Management, Total Quality Management (TQM), or Kaizen have been proposed over the years (cf. [5]). However, the

F. Johannsen (✉)
Faculty of Computer Science, University of Applied Sciences Schmalkalden, Schmalkalden, Germany
e-mail: f.johannsen@hs-sm.de

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_6

trend is moving towards the construction of enterprise-adapted quality management methods that align with business objectives and are easy to handle for employees (cf. [4, 10]).

Thereby, the PQM discipline is constantly evolving due to technological progress and new paradigms of doing business, which require existing approaches to be extended or new methods to be developed (cf. [11, 12]). For instance, technologies in industry 4.0 environments (e.g., Virtual Reality (VR) or Cyber-Physical-Systems (CPS)) offer new opportunities for PQM, such as maintenance works by help of VR glasses, providing additional construction drawings for the worker [13]. At the same time, machine sensors create real-time data, which enables the design of smart services that make use of this data to increase the "customer value" (e.g., a predictive maintenance service) [14]. Furthermore, the issue of environmental sustainability is increasingly being taken into account these days as well, and accordingly, corresponding performance metrics need to be considered in PQM projects (e.g., [15, 16]).

However, the design of new quality management methods or the adaptation of existing ones does not keep pace with the above-described developments. Hence, standardized PQM methods to redesign business processes to become more ecologically sustainable are missing as well as approaches for improving smart services (e.g., [17, 18]). In this respect, conceptual modeling and metamodeling can not only help to create enterprise-adapted PQM methods for various settings (cf. [10]), but also to extend existing approaches like Six Sigma. On this occasion, conceptual modeling has proven to be a valuable means in documenting, sharing, and processing the results created in PQM initiatives (cf. [19, 20]). This is decisive, because the communication of incomplete results, due to insufficient documentation, is a severe weakness in many PQM efforts hampering project orchestration endeavors (cf. [21, 22]).

This paper provides examples of how conceptual modeling and metamodels can be used to enhance the current capabilities of PQM methods in response to the aforementioned social and technological developments. The paper is structured as follows: In the next section, theoretical foundations are presented. Following that, metamodels are introduced for the topics of "smart services" and "environmental sustainability" alike. Different levels of granularity are presented for constructing corresponding metamodels. After a discussion, the paper concludes with a summary and an outlook.

2 Foundations

2.1 Process-Oriented Quality Management & Quality Techniques

Process-oriented quality management (PQM) deals with the improvement of business processes to fulfill the expectations of customers as well as stakeholders and essentially comprises the following activities (cf. [23, 24]): definition of customers/stakeholders and their needs, development of product and service features, redesign and improvement of processes to realize the defined features and anchoring the measures at the operational level [23]. In this regard, the creation of error-free products and processes by the workforce is expected to result in higher profits [25].

Considering this, various step-by-step approaches to improve business processes have been proposed over the decades (e.g., [26]). A well-known approach is the five-step procedure of Harrington [27], who suggests to focus on a company's critical processes when choosing a project. This method is referenced by Adesola and Baines [28], who propose a seven-step approach to improve business processes. Thereby, the operational character of the method to guide users in finding opportunities for improvement is emphasized (cf. [28]). In addition, Vakola and Rezgui [29] present an improvement method called "Condor methodology", which puts organizational issues and humans in the center of attention. However, Zellner [26] points out that many existing step-by-step methods suffer from a lack of methodological support at certain project stages.

Another strategy is the use of business process patterns to uncover known and recurring process weaknesses (e.g., [30, 31]). An overview of corresponding patterns is given by Fellmann et al. [30] for instance. In addition, process mining is seen as a valuable means to analyze as-is processes and arrive at should-be processes by help of event logs (cf. [32, 33]).

Against this background, quality techniques (e.g., Cause-and-Effect Diagram, etc.) (cf. [34]) are an essential component of any PQM method, because they serve to fulfill specific tasks in a project, e.g., the mapping of the process or the definition of key performance indicators (KPIs) [35, 36]. The literature proposes a wide variety of techniques (e.g., [34]), some of which have a more formal character (e.g., model types), while others (e.g., brainstorming) use less formal forms of knowledge representation, such as sketches or lists (e.g., [37]). A well-known collection of quality techniques is the so-called "7×7 toolbox" of the Six Sigma approach, which classifies techniques as "design techniques", "management techniques" or "customer techniques" among others [38]. In that context, recent research outlined beneficial synergies between quality techniques (cf. [39]). An exemplary overview of techniques is available at: http://tinyurl.com/mp7nmhk5

2.2 Operational Challenges of Process-Oriented Quality Management

A major challenge in PQM initiatives is to leverage employees' implicit process knowledge (cf. [40]) to purposefully develop process improvement suggestions *(challenge 1)*. Therefore, individuals possessing valuable insights into customer expectations and process weaknesses should actively participate in PQM projects. However, the workforce often lacks the time and resources to familiarize themselves with various methods for process improvement or to conduct several projects in parallel, which constitutes a second challenge [41] *(challenge 2)*. Third, if the selection of quality techniques does not align with the problem description and employees' skills, the project may fall short of expectations [42] *(challenge 3)*. Fourth, an improper documentation of project results not only hampers the project coordination but also increases the likelihood of duplicate work or the unheeded reversal of already achieved results (cf. [21]) *(challenge 4)*. Fifth, many existing PQM methods are considered oversized and too complex for projects with a narrow scope [43] *(challenge 5)*.

In addition to these operational challenges, there are also organizational and strategic issues (e.g., lack of a recognition culture) that may impede project success (cf. [42]). Though, the emphasis of this paper will be on the operational level.

2.3 Codification of Knowledge Via Conceptual Modeling

Conceptual modeling can help to resolve several of the aforementioned challenges (Sect. 2.2) through the purposeful codification of knowledge in PQM projects. Codification in this context involves not only converting human knowledge into machine-processable information [44], but also the structuring and representing the knowledge adequately, using tables, sketches or drawing among others [37, 45, 46].

In this regard, capturing results in form of conceptual models enables clear communication, preventing duplicate work and facilitating the coordination of parallel projects *(see challenge 4)*. Furthermore, conceptual models contribute to reducing complexity (e.g., [47]) by linking relevant pieces of information and visualizing obscured relationships, such as the link between customer requirements and process performance. This improves comprehensibility for quality issues allowing even employees with less pronounced PQM skills or experience to participate in projects *(see challenge 1)*. Moreover, appropriate model types that structure and interconnect information facilitate the efficient development of results, which saves resources and time (cf. [19]) *(see challenge 2)*. Recently proposed model types for PQM, along with approaches for their purposeful selection and integration (cf. [10]), simplify the choice of the most suitable techniques for a project *(see challenge 3)*. In this line, also the construction of enterprise-adapted PQM methods for initiatives of different scope is enabled (cf. [10]) *(see challenge 5)*. For an overview of quality techniques designed as conceptual model types refer to Johannsen and Fill [20] for instance.

3 Further Development of PQM for Emerging Topics with the Help of Conceptual Modeling

Emerging industry topics like "smart services" or the "environmentally sustainable" design of business processes (cf. [48, 49]) require PQM methods to be further developed. Hence, new concepts like environmental performance metrics (cf. [16]) or smart service quality dimensions (cf. [50]) have to be acknowledged and integrated with existing methods. This integration of new ideas can ideally be done at a metamodel level, whereby different granularity levels may be addressed (see Fig. 1). These can either be attributes of classes (level 3), the classes and relations themselves (level 2) or model types (level 1) (cf. [51]).

In this section, we present examples for extending existing PQM methods or techniques at the metamodel level to encompass the domains of "smart services" and "environmental sustainability".

3.1 Smart Services and Quality Management

Smart services represent digital services that are attributable to an increased digital interconnectivity of devices and machines, whereby sensors collect data to be processed via the internet or other networks [52]. Smart services come in various forms such as "smart production services", "smart transport and mobility services" or "smart healthcare services" to mention a few [53]. An example for a smart production service for elevators is the service "MAX" by "thyssenkrupp", wherein operational elevator data is sent to the cloud, analyzed and predictive maintenance works get triggered [53].

Smart services are inherently complex because they are composed of "physically delivered services", "digital services" and "physical elements" (sensors or devices)

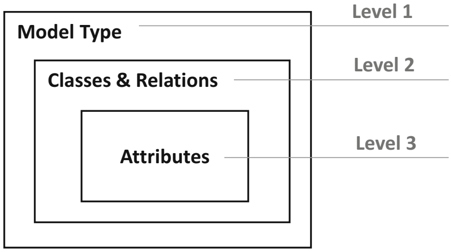


Fig. 1 Granularity levels [51]

[50]. In this respect, quality dimensions for physical products, traditional services but also digital services have been proposed in the literature (cf. [54–56]). Nevertheless, integrating these different perspectives is necessary to purposefully improve the quality of a smart service. Therefore, an integrated smart service quality framework for senior care services was proposed by Neuhuettler et al. [50]. Simultaneously, methods to operationally improve smart services in light of such frameworks, addressing product, service, and digital quality equally, are largely missing so far. Moreover, due to the novelty of the discipline (e.g., [57]), it remains unclear to what extent existing PQM methods can be modified to effectively work for smart service settings.

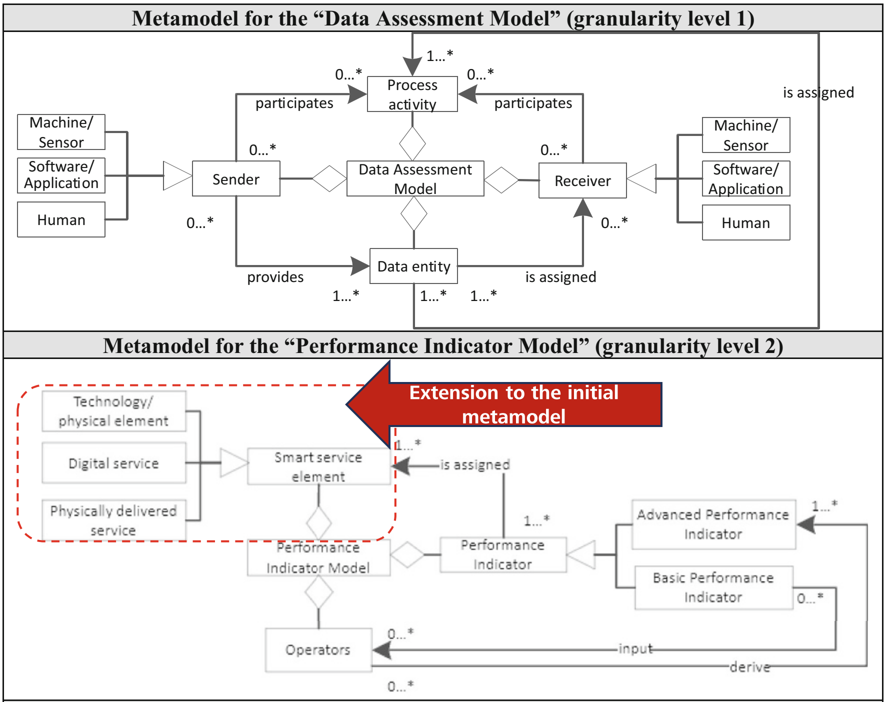
3.2 Conceptual Modelling for Improving Smart Production Services

In a recent work by Johannsen and Leist [17] a modeling tool-based Six Sigma approach for smart production services is introduced. In this context, 17 model types covering all stages of the Six Sigma cycle (DMAIC—Define, Measure, Analyze, Improve, Control) (e.g., [58]) were developed, to codify and structure emerging knowledge. For this purpose, the core concepts of established quality techniques were identified and converted into classes and relations of corresponding metamodels (cf. [17]). These propositions are subsequently referenced, to demonstrate the design of metamodels for PQM initiatives affecting smart service quality.

Considering **granularity level 1** (Fig. 1), a new model type called "Data Assessment Model" was introduced among others (cf. [17]). The model type outlines the participants in a smart service setting that receive and send data (receiver and sender). Thereby, the sender and receiver can be further specified as a "machine/sensor", an "application/software" or a "human". Furthermore, the relationships between the data and the process steps become evident. Additional attributes (e.g., topicality, consistency, etc.) allow for rating the data quality (e.g., [59]). The corresponding metamodel is shown in Fig. 2.

The metamodel for the "Performance Indicator Model" for smart production services is an example for the adaptation of an already existing metamodel on **granularity level 2** (see Fig. 2). Primarily, this model type helps to specify and operationalize KPIs in a project. Therefore, the KPIs defined in a model instance get structured and described in-depth. This way, the as-is performance measurement of a smart production service is enabled. An initial proposition for the "Performance Indicator Model" is described in Johannsen and Fill [20].

However, this metamodel needed to be adapted to work properly for smart production services. Hence, a class called "Smart service element" was added, along with specializations derived from the suggestions of Neuhuettler et al. [50] (e.g., physically delivered service, etc.). Consequently, the modeler can relate KPIs to the



Attributes of the "Affinity Diagram Model" (granularity level 3) in FDMM

(1) $MT_{AffinityModel} = \langle O^T_{AffinityModel}, D^T_{AffinityModel}, A_{AffinityModel} \rangle$
...

(2) $A_{AffinityModel}$ = {Name, Description, Pay-Off, Priority, Smart Service Element, Quality Dimension – physically delivered service, Quality Dimension – digital service – incubative dimension, Quality Dimension – digital service – active dimension, Quality Dimension – physical component & technology, Quality Dimension – data quality}
...

(3) $D^T_{AffinityModel}$ = {String, $\mathbf{Enum}_{\text{Quality Dimension – physically delivered service}}$, ...}
...

(4) $\mathbf{Enum}_{\text{Quality Dimension – physically delivered service}}$ = {Reliability, Responsiveness, Assurance, Empathy, Tangibles}
...

Fig. 2 Examples for further developments with respect to smart production services

elements of a smart service at an instance level and ensure that all smart service components are equally subjected to performance measurements in this way.

With respect to **granularity level 3** (Fig. 1), the "Affinity Diagram Model" (cf. [20]), which is a model type to support the development and structuring of improvement ideas is mentioned hereafter. In this model type, the modeling construct "solution" is used to specify improvement suggestions to overcome weaknesses of a smart production service.

To further define, which components of a smart service, along with corresponding quality dimensions, are affected by the improvement suggestions, attributes like *"Smart Service Element"*, *"Quality Dimension—physically delivered service"* or

"Quality Dimension—digital service—incubative dimension" (cf. [60]) were added to the "Solution" class. The quality dimensions and attribute values were derived from research about product quality (cf. [54]), service quality (cf. [61]), data quality (cf. [59]), and digital quality (cf. [60]). An excerpt from the corresponding FDMM presentation (Formalism for Describing ADOxx Meta Models and Models) (cf. [62]) is shown in Fig. 2. Accordingly, the excerpt presents the attributes and data types occurring in the "Affinity Diagram Model".

Figure 3 presents examples of the corresponding model types whereby the case of a predictive elevator maintenance service is used (e.g., [53]). The model types were implemented via the ADOxx metamodeling platform *(*adoxx.org*)*. In the "Data Assessment Model" instance, two activities "analyze cloud data" and "decide about solvability" are shown together with the processed data and the associated data sources. In the instance of the "Performance Indicator Model", KPIs to measure the smart service performance are defined. Finally, solutions to overcome the reasons for insufficient performance are collected (e.g., training of employees) and classified with the help of the "Affinity Diagram Model" (e.g., solutions affecting employees, sensors, etc.).

3.3 Environmental Sustainability and Quality Management

In view of the climate targets defined by the EU and the increasing awareness of environmental sustainability, more and more companies are searching for ways to reduce greenhouse gas emissions, energy consumption and resource expenditure [63]. In this context, PQM helps transform as-is procedures into more eco-friendly should-be processes [9]. Therefore, adapted PQM methods, such as "Green (Lean) Six Sigma" or "Green Lean Management", have been proposed in the recent past (e.g., [18]). However, there are no accepted standards yet because the research field is rather new, and there is fuzziness about the term "green quality" [64].

In literature, a variety of quality techniques are proposed for improving business processes with respect to environmental sustainability. These suggestions often represent adaptations or enhancements of well-established techniques (e.g., Value-Stream-Map) (cf. [65]), whereby these modifications and further developments can be systematically described at a metamodel level as well.

3.4 Conceptual Modelling for Environmental Sustainability

To arrive at metamodels for environmental sustainability, we enhance model types that were developed in our prior research on modeling-based PQM independent of a particular domain (cf. [20]).

With respect to **granularity level 2** (Fig. 1), we exemplarily present an extension of the *CTQ-/CTB-Matrix* (cf. [34]) by the classes "Voice of the Environment

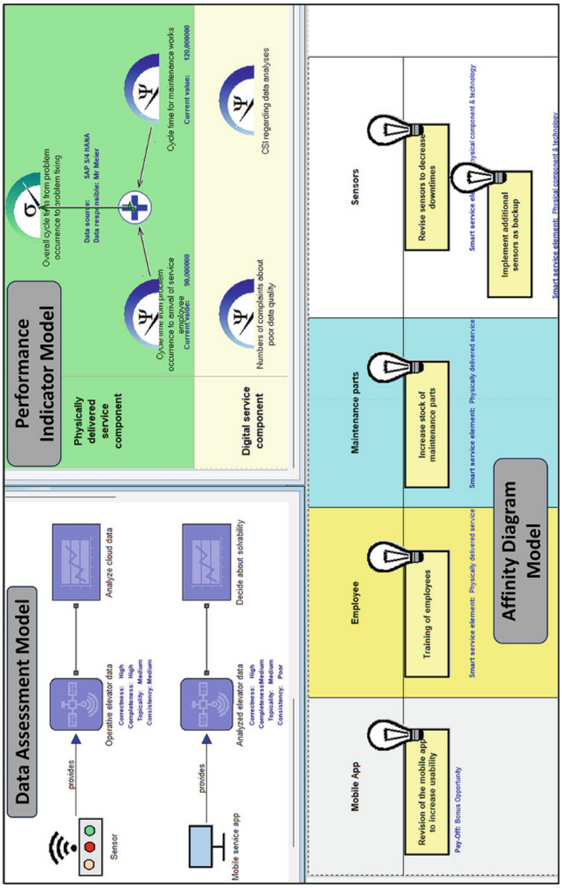


Fig. 3 Screenshots of instances of the “Data Assessment Model”, “Performance Indicator Model” and “Affinity Diagram Model”

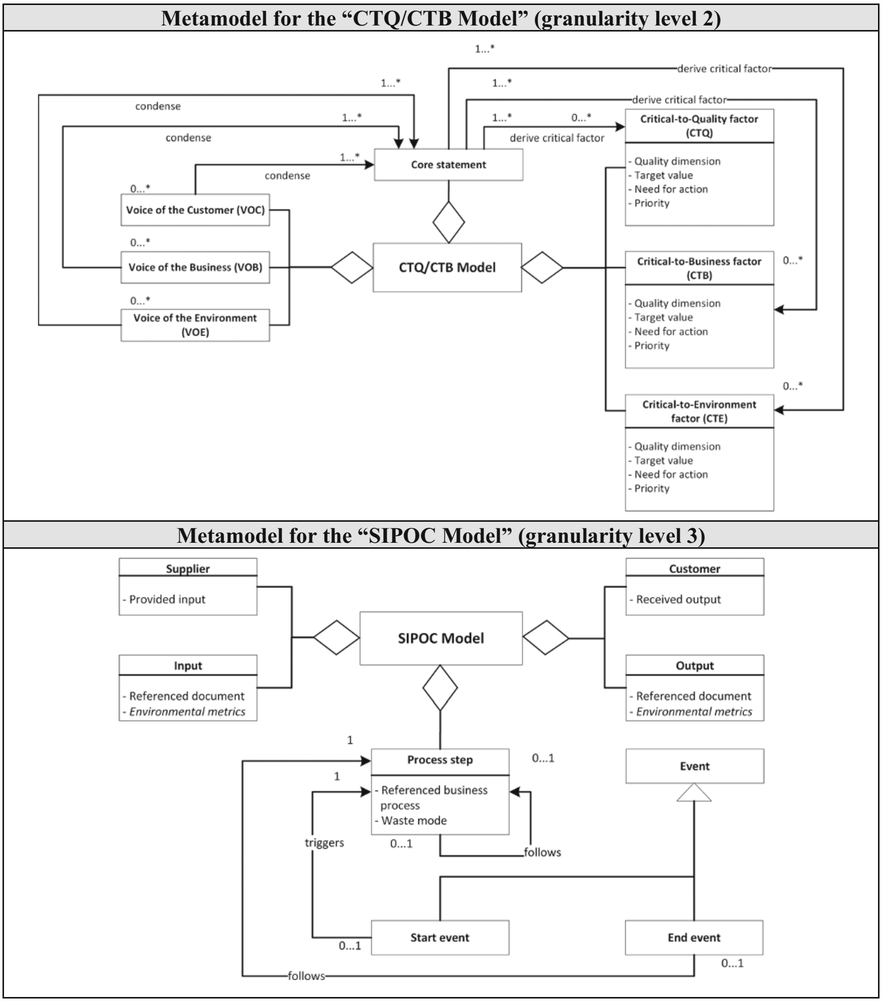


Fig. 4 Examples for further developments with respect to environmental sustainability

(VOE)" [66] and "Critical-to-Environment (CTE)" factor at a metamodel level. Generally, the *CTQ-/CTB-Matrix* helps to collect and structure customer (Voice of the Customer—VOC) and stakeholder (Voice of the Business—VOB) requirements alike to derive project goals, i.e., "Critical-to-Quality (CTQ)" and "Critical-to-Business (CTB)" factors. These concepts are now enhanced by environmental requirements (VOEs) (cf. [66]), enabling the specification of environmental goals (CTE factors) (see Fig. 4). In this way, environmental objectives can also be considered when defining PQM projects. An instance is shown in Fig. 5. Hence, based on three VOC statements, a CTQ and a CTE factor are derived, determining

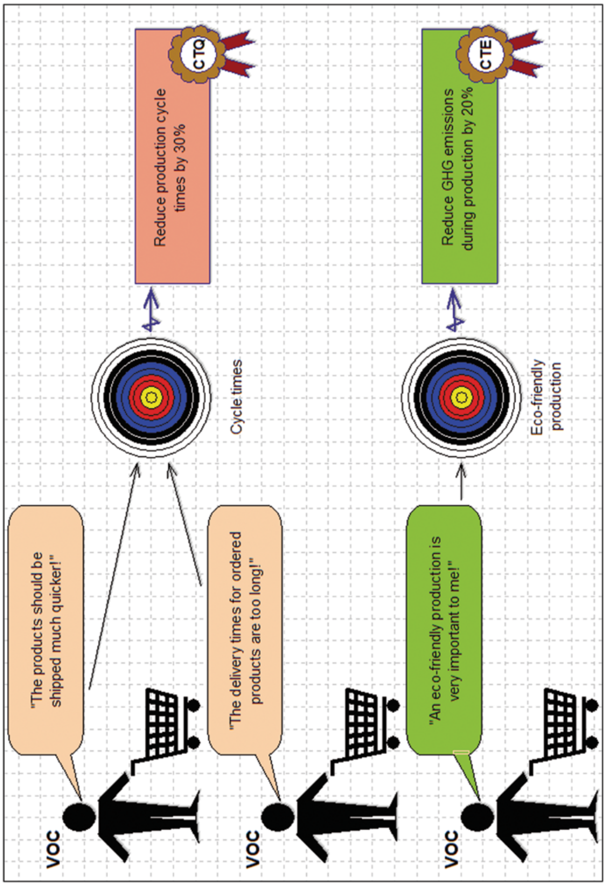


Fig. 5 Example for the CTQ/CTB-Model considering environmental sustainability

the project goals. Thereby, the reduction of greenhouse gas emissions during production is defined as a quality goal for instance.

With regard to **granularity level 3**, the *SIPOC Diagram* should serve as an example. Essentially, this model type provides a brief overview of a business process to be improved including the input processed, the output created as well as the suppliers and customers involved (cf. [34]). Figure 4 shows the proposed adapted metamodel of the SIPOC Diagram (SIPOC Model). Thereby, information about the input processed and output produced is enriched by environmental performance metrics like “emissions to air” or “solid waste” (cf. [67]). These performance metrics are integrated as attributes of the classes “Input” and “Output”. This way, the ecological footprint during the delivery of materials or the production of goods can be recorded in a model instance. Additionally, the so-called “waste mode” (e.g., overproduction, unnecessary transport, etc.) [68] is proposed as an attribute for the class “Process step”. This information indicates potentially negative effects on the environment during process execution.

It should be mentioned that research on PQM for environmental sustainability is a novel discipline, and hence, new model types to support environmental PQM initiatives are largely missing thus far. Rather, the current discussion revolves around the extent to which existing quality techniques can be purposefully used to achieve environmental goals. Accordingly, examples for granularity level 1 (Fig. 1) are to be developed in upcoming steps.

4 Discussion

With the rise of new technologies and novel ways of doing business, existing PQM approaches have to be constantly further developed (cf. [11]). Conceptual modeling not only helps address many of the challenges in PQM projects as outlined in Sects. 2.2 and 2.3 but also serves as a starting point for adapting established methods and techniques to function in new environments.

Metamodeling, therefore, supports the explication of the functionality and core concepts of quality techniques through the use of classes and relations. New requirements, such as those arising from the emergence of smart services or the increased awareness of environmental sustainability, can be integrated at the metamodel level, as demonstrated in Sect. 3. Consequently, the functionality of quality techniques can be extended or adapted, leading to the creation of new PQM methods for various domains. In this context, Agile Modeling Method Engineering (AMME) [69] assists in combining the newly created metamodels to develop PQM methods that best match with the project situation.

By using metamodeling platforms like ADOxx, prototypes supporting the agile construction of adapted PQM methods can be implemented (cf. [10]). During method construction, knowledge about the valuable combination of quality techniques to utilize synergies is crucial (cf. [70]). An example would be the further processing of quality goals (CTQs, etc.), which can be defined by help of the

CTQ-/CTB-Matrix via the *Measurement Matrix* (cf. [34]) to derive suitable KPIs. Accordingly, the beneficial interplay between quality techniques is analyzed in Johannsen [70], and indicators for synergies are outlined at the metamodel level.

In summary, this paper demonstrates that PQM can be purposefully further developed, and the boundaries of the discipline can be shifted through conceptual modeling. It becomes evident that concepts put forth by business trends can be integrated with metamodels at different levels of granularity. This includes the addition of attributes, the introduction of classes or the development of new model types. However, the modeling method engineer must decide on the most promising option.

5 Outlook

The paper demonstrates opportunities to refine the PQM discipline through conceptual modeling and metamodels. For this purpose, examples from the areas of "smart services" and "environmental sustainability" were presented. Both research fields are highly dynamic, and universally accepted PQM standards do not yet exist for these. The artifacts in this paper therefore represent research-in-process results.

Considering this, a comprising evaluation of the introduced model types is to be performed in upcoming steps. Furthermore, ADOxx will be used to create prototypes that facilitate the easy construction of enterprise-adapted methods for PQM. The prototypes realized in this manner will undergo usability studies to revise and improve them for practical application.

References

1. Hecker Consulting: Die 7 größten Herausforderungen für Unternehmen im Jahr 2023. https://www.hco.de/blog/die-7-grossten-herausforderungen-fur-unternehmen-im-jahr-2023. Accessed 22 Dec 2023
2. Statista: Gaspreise für Gewerbe- und Industriekunden in Deutschland in den Jahren 2012 bis 2022. https://de.statista.com/statistik/daten/studie/168528/umfrage/gaspreise-fuer-gewerbe-und-industriekunden-seit-2006/. Accessed 22 Dec 2023
3. Statista: In welchen Bereichen in Ihrem Unternehmen sehen Sie in den kommenden zwölf Monaten die größten Herausforderungen? https://de.statista.com/statistik/daten/studie/1297695/umfrage/umfrage-zu-den-groessten-herausforderungen-von-deutschen-unternehmen/. Accessed 22 Dec 2023
4. Clochet, A.: PEX report 2023 - helping organizations master process improvements to remain competitive in hybrid-work environments (2022)
5. Stracke, C.: Process-oriented quality management. In: Ehlers, U.-D., Pawlowski, J.M. (eds.) Handbook on Quality and Standardisation in E-Learning, pp. 79–96. Springer, Berlin (2006)
6. Leist, S., Karagiannis, D., Johannsen, F., Penzel, H.-G.: From business engineering to digital engineering: the role of metamodeling in digital transformation. In: Aier, S., Rohner, P., Schelp, J. (eds.) Engineering the Transformation of the Enterprise: A Design Science Research Perspective, pp. 51–71. Springer, Cham (2021)

7. Pande, P., Neuman, R., Cavanagh, R.: The Six Sigma Way: How to Maximize the Impact of your Change and Improvement Efforts. McGraw Hill Professional, New York (2014)
8. Harmon, P.: Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals. Elsevier, Oxford (2007)
9. Delgadillo, R., Medini, K., Wuest, T.: A DMAIC framework to improve quality and sustainability in additive manufacturing—a case study. Sustainability. **14**, 581 (2022)
10. Johannsen, F.: Situational roadmap development for business process improvement via a modeling tool. In: 31st European Conference on Information Systems (ECIS 2023), Kristiansand (2023)
11. Foidl, H., Felderer, M.: Research challenges of industry 4.0 for quality management. In: Felderer, M., Piazolo, F., Ortner, W., Brehm, L., Hof, H.-J. (eds.) Innovations in Enterprise Information Systems Management and Engineering, pp. 121–137. Springer, Cham (2016)
12. Sony, M., Antony, J., Douglas, J.A.: Essential ingredients for the implementation of Quality 4.0. TQM J. **32**, 779–793 (2020)
13. Hänisch, T.: Grundlagen Industrie 4.0. In: Andelfinger, V.P., Hänisch, T. (eds.) Industrie 4.0, pp. 9–31. Springer, Wiesbaden (2017)
14. Pöppelbuß, J., Durst, C.: Smart Service Canvas—Ein Werkzeug zur strukturierten Beschreibung und Entwicklung von Smart-Service-Geschäftsmodellen. In: Dienstleistungen 4.0: Geschäftsmodelle-Wertschöpfung-Transformation. Band 2. Forum Dienstleistungsmanagement, pp. 91–110. Springer, Wiesbaden (2017)
15. Farrukh, A., Mathrani, S., Sajjad, A.: A natural resource and institutional theory-based view of green-lean-six sigma drivers for environmental management. Bus. Strateg. Environ. **31**, 1074–1090 (2022)
16. Gholami, H., Jamil, N., Mat Saman, M.Z., Streimikiene, D., Sharif, S., Zakuan, N.: The application of green lean six sigma. Bus. Strateg. Environ. **30**, 1913–1931 (2021)
17. Johannsen, F., Leist, S.: Six sigma for smart production services–towards a modeling tool-based approach. In: 29th European Conference on Information Systems. ECIS, Marrakech, Morocco (2021)
18. Farrukh, A., Mathrani, S., Taskin, N.: Investigating the theoretical constructs of a green lean six sigma approach towards environmental sustainability: a systematic literature review and future directions. Sustainability. **12**, 8247 (2020)
19. Johannsen, F., Mang, F., Fill, H.-G., Hofmann, S.: Tool-based codification in business process improvement and the impact on problem-solving. Enterp. Model. Inf. Syst. Archit. **18**, 1–30 (2023)
20. Johannsen, F., Fill, H.-G.: Meta modeling for business process improvement. Bus. Inf. Syst. Eng. **59**, 251–275 (2017)
21. Samson, D., Challis, D.: Patterns of business excellence. Meas. Bus. Excell. **6**, 15–21 (2002)
22. Breyfogle, F.W.: Process improvement projects shortcomings and resolution. Int. J. Lean Six Sigma. **1**, 92–98 (2010)
23. Nedeliaková, E., Panák, M.: New trends in process-oriented quality management. Procedia Econ. Financ. **34**, 172–179 (2015)
24. Juran, J.M., De Feo, J.A.: Juran's Quality Handbook: The Complete Guide to Performance Excellence. McGraw-Hill Education (2010)
25. Töpfer, A.: Six Sigma als Projektmanagement für höhere Kundenzufriedenheit und bessere Unternehmensergebnisse. In: Töpfer, A. (ed.) Six Sigma - Konzeption und Erfolgsbeispiele für praktizierte Null-Fehler-Qualität, pp. 45–99. Springer, Berlin (2007)
26. Zellner, G.: A structured evaluation of business process improvement approaches. Bus. Process. Manag. J. **17**, 203–237 (2011)
27. Harrington, J.: Business Process Improvement—The Breakthrough Strategy for Total Quality, Productivity and Competiveness. McGraw Hill, New York (1991)
28. Adesola, S., Baines, T.: Developing and evaluating a methodology for business process improvement. Bus. Process. Manag. J. **11**, 37–46 (2005)

29. Vakola, M., Rezgui, Y.: Critique of existing business process re-engineering methodologies: the development and implementation of a new methodology. Bus. Process. Manag. J. **6**, 238–250 (2000)
30. Fellmann, M., Koschmider, A., Laue, R., Schoknecht, A., Vetter, A.: Business process model patterns-classification and repository of published work. In: EMISA Forum, vol. 1, pp. 67–75. De Gruyter (2019)
31. Koschmider, A., Laue, R., Fellmann, M.: Business process model anti-patterns: a bibliography and taxonomy of published work. In: 27th European Conference on Information Systems (ECIS 2019), Stockholm (2019)
32. Satyal, S., Weber, I., Paik, H.-Y., Di Ciccio, C., Mendling, J.: Business process improvement with the AB-BPM methodology. Inf. Syst. **84**, 283–298 (2019)
33. Van Der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011)
34. Meran, R., John, A., Roenpage, O., Staudter, C.: Six Sigma+Lean Toolset. Springer, Berlin (2013)
35. McQuater, R.E., Scurr, C.H., Dale, B.G., Hillmann, P.G.: Using quality tools and techniques successfully. TQM Mag. **7**, 37–42 (1995)
36. Johannsen, F.: State of the art concerning the integration of methods and techniques in quality management—literature review and an agenda for research. In: 19th European Conference on Information Systems (ECIS 2011), Helsinki (2011)
37. Anaby-Tavor, A., Amid, D., Fisher, A., Bercovici, A., Ossher, H., Callery, M., Desmond, M., Krasikov, S., Simmonds, I.: Insights into enterprise conceptual modeling. Data Knowl. Eng. **69**, 1302–1318 (2010)
38. Magnusson, K., Kroslid, D., Bergman, B.: Six Sigma the Pragmatic Approach. Professional Public Service (2004)
39. Johannsen, F., Zellner, G., Griesberger, P.: Creating a functional interdependency map for supporting the "act of improvement" in business process improvement projects. Int. J. Serv. Sci. Manag. Eng. Technol. **13**, 1–23 (2022)
40. Seethamraju, R., Marjanovic, O.: Role of process knowledge in business process improvement methodology: a case study. Bus. Process. Manag. J. **15**, 920–936 (2009)
41. Gijo, E.V., Rao, T.: S.: six sigma implementation—hurdles and more hurdles. Total Qual. Manag. Bus. Excell. **16**, 721–725 (2005)
42. Antony, J., Gupta, S.: Top ten reasons for process improvement project failures. Int. J. Lean Six Sigma. **10**, 367–374 (2019)
43. Davis, D.: 3rd Biennial PEX Network Report: State of the Industry - Trends and Success Factors in Business Process Excellence (2013)
44. Bork, D., Fill, H.-G.: Formal aspects of enterprise modeling methods: a comparison framework. In: 47th Hawaii International Conference on System Sciences (HICSS 2014), pp. 3400–3409. IEEE, Waikoloa, Hawaii (2014)
45. Wand, Y., Weber, R.: Research commentary: information systems and conceptual modeling - a research agenda. Inf. Syst. Res. **13**, 363–376 (2002)
46. Hall, M.: Knowledge management and the limits of knowledge codification. J. Knowl. Manag. **10**, 117–126 (2006)
47. Recker, J.C., Lukyanenko, R., Jabbari Sabegh, M., Samuel, B., Castellanos, A.: From representation to mediation: a new agenda for conceptual modeling research in a digital world. MIS Q. Manag. Inf. Syst. **45**, 269–300 (2021)
48. Houy, C., Reiter, M., Fettke, P., Loos, P., Hoesch-Klohe, K., Ghose, A.: Advancing business process technology for humanity: opportunities and challenges of green BPM for sustainable business activities. In: Green Business Process Management: Towards the Sustainable Enterprise, pp. 75–92. Springer (2012)
49. Beverungen, D., Müller, O., Matzner, M., Mendling, J., Vom Brocke, J.: Conceptualizing smart service systems. Electron. Mark. **29**, 7–18 (2019)

50. Neuhuettler, J., Ganz, W., Liu, J.: An integrated approach for measuring and managing quality of smart senior care services. In: Ahram, T.Z., Karwowski, W. (eds.) Advances in the Human Side of Service Engineering, pp. 309–318. Springer, Cham, Switzerland (2017)
51. Fill, H.-G., Johannsen, F.: A knowledge perspective on big data by joining Enterprise modeling and data analyses. In: 49th Hawaii International Conference on System Sciences (HICSS 2016), pp. 4052–4061. IEEE, Kauai, Hawaii (2016)
52. Pöppelbuß, J.: Smart service. In: Enzyklopaedie der Wirtschaftsinformatik - Online Lexikon. https://wi-lex.de/index.php/lexikon/inner-und-ueberbetriebliche-informationssysteme/sektorspezifische-anwendungssysteme/smart-service/. Accessed 22 Dec 2023
53. Acatech: Smart Service Welt: Digitale Serviceplattformen—Praxiserfahrungen aus der Industrie—Best Practices (2016)
54. Garvin, D.: What does product quality really mean? Sloan Manag. Rev. **26**, 25–43 (1984)
55. Cristobal, E., Flavián, C., Guinaliu, M.: Perceived e-service quality (PeSQ) measurement validation and effects on consumer satisfaction and web site loyalty. Manag. Serv. Qual. **17**, 317–340 (2007)
56. Parasuraman, A., Zeithaml, V., Berry, L.: A conceptual model of service quality and its implications for future research. J. Mark. **49**, 41–50 (1985)
57. Nenadál, J.: The new EFQM model: what is really new and could be considered as a suitable tool with respect to quality 4.0 concept? Qual. Innov. Prosper. **24**, 17–28 (2020)
58. Snee, R., Hoerl, R.: Leading Six Sigma. Prentice Hall, New York (2003)
59. Klier, M.: Metriken zur Bewertung der Datenqualität—Konzeption und praktischer Nutzen. Inform. Spektrum. **31**, 223–236 (2008)
60. Santos, J.: E-service quality: a model of virtual service quality dimensions. Manag. Serv. Qual. **13**, 233–246 (2003)
61. Parasuraman, A., Zeithaml, V., Berry, L.: SERVQUAL: a multiple-item scale for measuring consumer perception of service quality. J. Retail. **64**, 12–40 (1988)
62. Fill, H.-G., Redmond, T., Karagiannis, D.: FDMM: a formalism for describing ADOxx meta models and models. In: 14th International Conference on Enterprise Information Systems (ICEIS 2012), pp. 133–144. ICEIS, Poland, Wroclaw (2012)
63. Belousova, V., Bondarenko, O., Chichkanov, N., Lebedev, D., Miles, I.: Coping with greenhouse gas emissions: insights from digital business services. Energies. **15**, 2745 (2022)
64. Gouda, S.K., Awasthy, P., Krishnan, T., Sreedevi, R.: What does "green quality" really mean? TQM J. **31**, 52–69 (2018)
65. Banawi, A.-A., Bilec, M.: Applying lean, green, and six-sigma framework to improve exterior construction process in Saudi Arabia. J. Constr. Eng. Project Manag. **4**, 12–22 (2014)
66. Trappey, A.J., Ou, J.J., Lin, G.Y., Chen, M.-Y.: An eco-and inno-product design system applying integrated and intelligent QFDE and TRIZ methodology. J. Syst. Sci. Syst. Eng. **20**, 443–459 (2011)
67. Cherrafi, A., Elfezazi, S., Govindan, K., Garza-Reyes, J.A., Benhida, K., Mokhlis, A.: A framework for the integration of green and lean six sigma for superior sustainability performance. Int. J. Prod. Res. **55**, 4481–4515 (2017)
68. de Souza, R.V.B., Carpinetti, L.C.R.: A FMEA-based approach to prioritize waste reduction in lean implementation. Int. J. Qual. Reliab. Manag. **31**, 346–366 (2014)
69. Karagiannis, D.: Conceptual modelling methods: the AMME agile engineering approach. In: Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W. (eds.) Domain-Specific Conceptual Modeling, pp. 3–19. Springer, Cham (2018)
70. Johannsen, F.: Functional interdependencies between quality techniques reverting to meta models. In: 13th Internationale Tagung Wirtschaftsinformatik (WI 2017). University of St. Gallen, St. Gallen, Switzerland (2017)

Metamodeling Platforms: Observations and Evolutions

Harald Kühn

Abstract Metamodeling platforms are an established foundation for the realization of domain-specific modeling tools in the academic and research domain as well as in the industrial and business domain. This chapter illustrates observations made using metamodeling platforms in industry such as the "democratization of modeling" and the usage of method fragments in Agile Modeling Method Engineering (AMME). In recent years, metamodeling platforms evolved both due to their intensified use in practice and rapid technological developments in services they use. An enhanced meta-modeling platform architecture is presented, incorporating the notion of service-orientation. It is proposed to extend method engineering to method operation, introducing the Continuous Modeling Method Operation approach (CMMO). To combine the expressiveness of domain-specific languages (DSL) and the power of large language models (LLM), a system architecture and integration proposal is presented.

Keywords Metamodeling platforms · Agile modeling method engineering · Method fragments · Metamodeling platform architecture · Continuous modeling method operation · Integration of DSL and LLM · ADOxx

1 Introduction

Metamodeling platforms provide the foundation for flexible modeling method engineering and efficient modeling tool development [1–4]. They are widely used both in the academic and research domain as well as in the industrial and business domain [5–8]. Metamodeling platforms are defined as "*[. . .] software environments allowing the definition, usage and maintenance of a method's elements: (a) metamodels describing problem-specific modelling languages, (b) mechanisms &*

H. Kühn (✉)
BOC Products & Services AG, Vienna, Austria
e-mail: harald.kuehn@boc-group.com

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_7

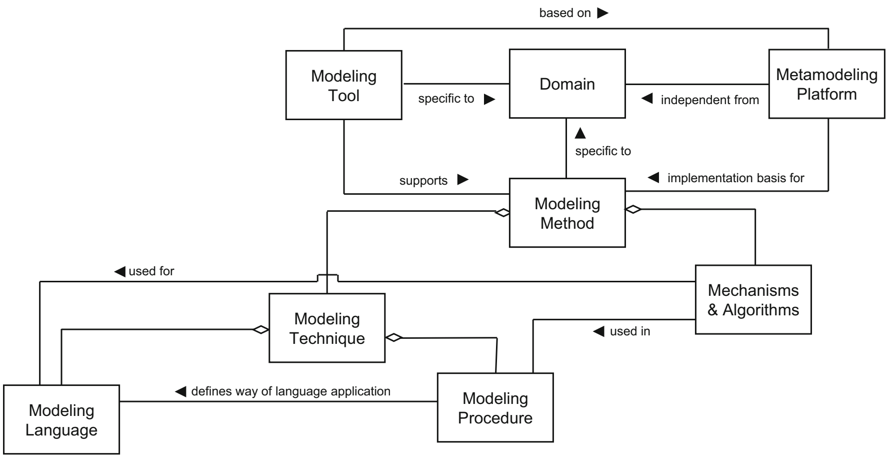


Fig. 1 Modeling methods and metamodeling platforms

algorithms working on models and their underlying metamodels, and (c) procedure models representing process descriptions how to apply the metamodels and the corresponding mechanisms [. . .]" [9, p. 215] (Fig. 1).

Situational Method Engineering (SME) [10, 11] considers the need, to tailor methods or method fragments to specific requirements of a particular situation or project. This approach is highly relevant in fields where standardized methods, standardized modeling languages and/or pre-defined mechanisms working on models do not cover the full scope of a situation's particularity and purpose. In such situations, domain-specific solutions create value in efficiency, quality and/or reduced costs. Examples are situational language extensions to BPMN to allow the representation of risks and controls for Internal Control Systems (ICS) or specific ESG[1] attributes to ArchiMate language elements to consider the CO_2 footprint, the energy consumption or the pollution contribution of a business process, an application component, an IT service or an infrastructure equipment [12]. Examples for situational mechanism extensions are report generations for company-specific risk and control reports or the cost calculation of CO_2 compensation measures based on specific ESG attributes of ArchiMate elements.

Agile Modeling Method Engineering (AMME) [13] is an approach that applies the principle of agility from Software Engineering to the practice of Modeling Method Engineering. AMME encompasses a lifecycle with five phases: Create, Design, Formalize, Develop, and Deploy/Validate. This approach is supported by tools like the modeling method domain-specific language MM-DSL and the ADOxx metamodeling platform [5, 14]. It focuses on evolving modeling methods iteratively based on changing requirements and feedback loops.

[1] ESG: Environmental, Social, Governance.

Modeling methods often originate initially from research and related publications. According to [1], a modeling method consist of three method elements: (1) a modeling language, (2) a modeling procedure and (3) mechanisms & algorithms. The combination of a modeling language together with its modeling procedure is called modeling technique. To evaluate and use modeling methods and its method elements, it is essential to have them implemented in software-based tools; processing capabilities require a digital representation. Domain-specific modeling tools can be created by using metamodeling platforms and applying approaches such as SME and AMME. Tools bring the "conceptual groundwork" from research and innovation into real-life applications to validate the applicability of the method. A tool allows to gather domain-relevant information by applying the method, to analyze this information and to provide results based on the method. Thus, the tool usage provides input for a method's usability evaluation, the validation of its results and recommendations to improve the method and its tooling.

In the context of the ADOxx metamodeling platform, a rich eco-system of domain-specific tools and related communities and experiences has been established over many years. This encompasses already more than 80 ADOxx-based tools from numerous international research institutions and various application domains which are freely available [15–17]. Followed by ADONIS and ADOIT Community Editions for BPM and EAM [18], which are freely available as well. And finally the industry-leading Management Office product suite of BOC Group with the BPM tool ADONIS, the EAM tool ADOIT and the GRC tool ADOGRC [19] accompanied by professional services such as Software-as-a-Service (SaaS), Managed Services, Training and Consulting.

The remainder of this chapter is structured as follows: Sect. 2 discusses various observations gathered from projects applying metamodeling platform technology and related domain-specific modeling tools, particularly based on ADOxx. In Sect. 3 important evolutions of metamodeling platforms are presented. These range from evolutions which are already on the way as well as evolutions which are expected to come in the near future. The chapter closes with an outlook to the potential directions of metamodeling platforms and related technologies.

2 Metamodeling Platforms: Observations

Based on the experiences applying metamodeling platforms to create domain-specific modeling tools since many years as well as from a broad range of projects in the above mentioned communities, the following observations have been made.

2.1 Democratization of Modeling

During the early adoption phase of metamodeling and domain-specific modeling tools, the stakeholders and users have mainly been technical experts, method engineers, business experts with IT background or IT professionals. Since then, conceptual models and their use in Enterprise Management has dramatically extended [20]. In almost every large or mid-sized organization, models such as process models, organizational diagrams, application landscapes, risk and control models etc. are used by a broad range of users. This includes not only the consumption of model information itself. It ranges from active modeling using conceptual models for descriptive and prescriptive tasks, via analytical and evaluative tasks down to tool configuration by end-users ("metamodeling"). This broad range of usage is called "Democratizing of Modeling". It is driven by several reasons that reflect the evolving needs of businesses, technological advancements, and changing workforce dynamics.

- Bridging the skills gap: modeling tools with user-friendly interfaces and guided workflows make it possible for non-technical staff to contribute effectively. These advancements have lowered the barriers to entry, making it possible for a wider range of users to engage with these tools. Organizations get more scalable, without the bottleneck of relying on a limited number of expert modelers.
- Rapid response to market changes: democratizing modeling enables organizations to leverage the collective knowledge and agility of their entire workforce, rather than relying solely on specialized teams for faster and more effective responses. This inclusivity leads to more well-rounded modeling results and informed decisions using the models, as insights are gathered from various perspectives within the organization.
- Enhanced collaboration and innovation: when more people are involved in the modeling process, it fosters collaboration and encourages a culture of innovation. Different departments and teams can contribute their unique insights, leading to more creative solutions and breakthrough ideas.
- Empowering employees: providing employees across various levels and functions with the tools to engage in modeling tasks can lead to greater job satisfaction and a sense of empowerment.
- Efficient resource utilization and cost-effectiveness: by enabling non-modeling-experts to handle certain modeling tasks, organizations can make better use of their human resources. It frees up specialized teams to focus on more complex challenges, thereby improving overall efficiency. Also the dependency to external and specialized consultants can be reduced.

In addition, improved contextualization features of metamodeling platforms and their related method elements, lead to a much higher degree of guidance and self-service in modeling tools (see Fig. 2). A context is described by the triple of an actor, role and persona, for instance a person (actor: "Human") reads a process dashboard (role: "Consumer") to view the approval state of the processes he/she

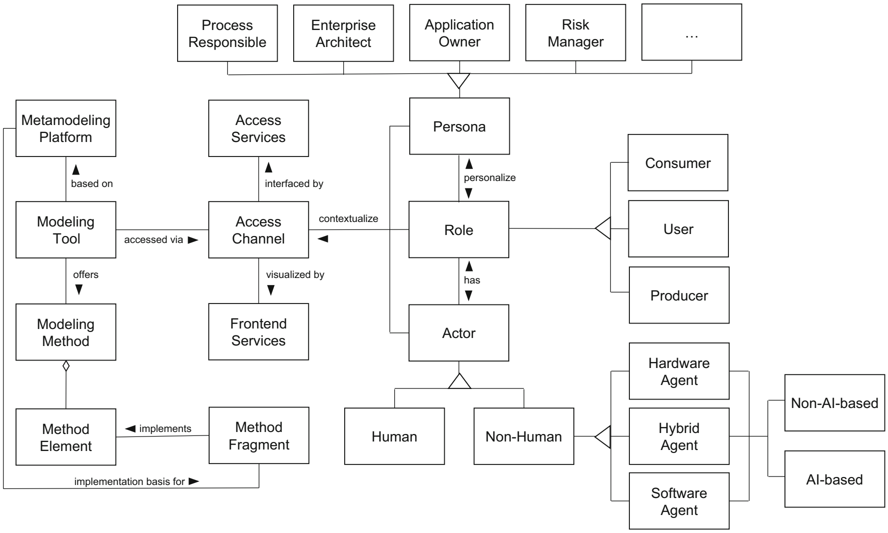


Fig. 2 Contextualization via persona, roles and actors

is responsible for (persona: "Process Responsible"). A modeling tool is accessible via various access channels. Access channels for non-technical users are typically represented by a group of widgets and user-interfaces in a specific context. Self-service refers to empowering such users to independently use access channels to access, create, manage, and utilize models and data without the need for specialized technical expertise. Features include guided processes, wizards, or templates that help users through the modeling process and intuitive and easy-to-navigate user interfaces. These user interfaces often include graphical elements, drag-and-drop functionalities, and a clear, non-technical language, enabling users with varied skill levels to effectively interact with the system.

During the self-service process, users act in a concrete "Role". As "Consumer", they rely on models, services, or results provided by the tool. As "User", they contribute via editors or specialized UI widgets to the model repository. As "Producer", they configure the tool components, metamodels of modeling languages and/or integration capabilities to provide the right tooling for consumers and users.

An entity fulfilling one of these roles is called "Actor". An Actor can be human or non-human, from an application point of view they are equivalent. This is of special interest during implementation and usage of domain-specific tools. Depending of the situational requirement of a modeling method's element, instead of a human person the related task could be executed by a machine or device (hardware agent), a digital agent (software agent) or a mixture of both (hybrid agent). For example, in a safety relevant situation where a human being could be harmed, a robot agent could take over the responsibility executing the relevant task of the method. Or in a business process optimization project the necessary process assessment could be taken over by an AI-based software agent analyzing the relevant processes and providing assessment results and optimization proposals, supporting or even replacing a business process consultant for this dedicated task.

A “Persona” represents a fictional but well-defined user group with distinct characteristics, needs, behaviors, and goals. Personas are typically identified based on user research and serve as archetypal user representations. Examples are process responsibles, enterprise architects, application owners or risk managers. Together with “Role” and “Actor”, “Persona” is the third element to contextualize the usage of a metamodeling platform component via distinct access channels. Recognizing non-human actors (like AI agents, software bots, and hardware systems) expands the scope of modeling beyond human-centric processes. This inclusion allows for automated, AI-driven analysis and modeling, which can handle complex tasks, process large model bases, and provide insights that might not be evident to human modelers.

The dynamic usage of metamodeling platform components by both human actors and non-human actors led to open metamodeling platform architectures with a variety of access channels. On the technology level a rich set of APIs, data exchange formats, connectors and REST endpoints are typically offered (“Access Services”). On the UI level often the possibility to integrate the whole UI or parts of it via widgets into the UI of external systems is provided (“Frontend Services”).

2.2 Situational Agility Using Method Fragments

Applying AMME incorporates the use of principles such as adaptability, extensibility, integrability, operability and usability [13]. This leads in the context of metamodeling platforms to the need, not only to support pre-defined modeling methods and their specific method elements, but also to allow their refinement, adaption, and extension. Or even to integrate complete new modeling languages, usage procedures and mechanisms and features on-the-fly into an existing modeling method and its related tooling. Trends as “Composable Enterprise” apply such an approach on enterprise scale [21]: “*A composable enterprise is an organization that can innovate and adapt to changing business needs through the assembly and combination of packaged business capabilities. Packaged business capabilities will be sourced from third parties or composed internally. They will deliver more unique and customized application experiences to application users.*”

In the domain of SME, concepts such as “method chunk” and “method fragment” have been established to enhance situational flexibility [10, 22]. There, the presented concepts focus either on the “product part” (modeling language) or the “process part” (modeling procedure). Application scenarios of these concepts in the domain of information systems interoperability can be found in [23, 24]. The implementation of tools need in many cases specific algorithmic extensions to analyze domain-specific models, simulate them or transform model structures from one language to another. Therefore, an additional dimension “mechanisms &

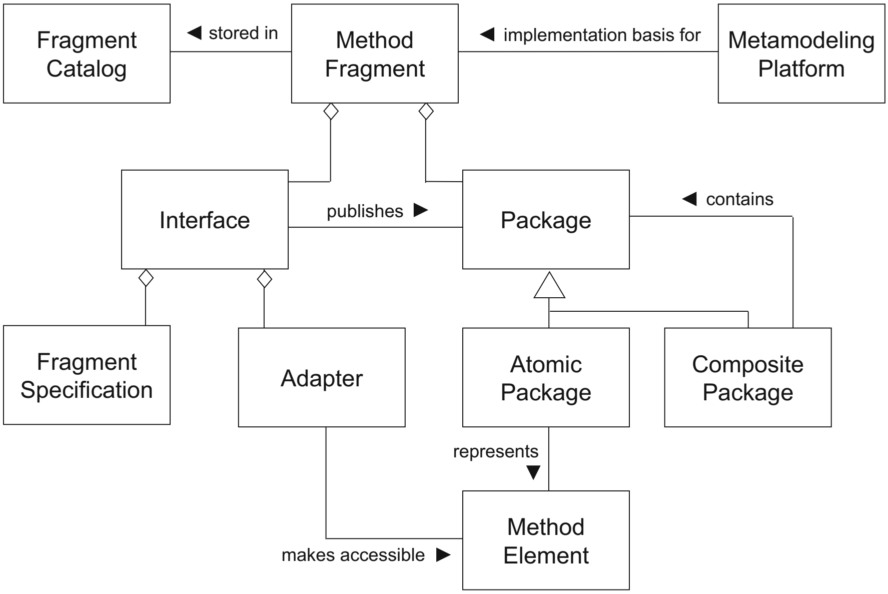


Fig. 3 Method fragments for agile modeling method engineering

algorithms" is needed. To address this requirement, an extended method fragment definition is proposed which incorporates all three modeling methods elements "modeling language", "modeling procedure" and "mechanisms & algorithms" [11].

A method fragment consists of an interface part and a package part (see Fig. 3). The package contains the actual feature set of a method fragment. For example, a package could be a set of modeling classes, relations and attributes representing a process modeling language or a procedure model how to describe standard operating procedures in the pharma industry or a specific algorithm to calculate the "value at risk" in an Enterprise Risk Management methodology. An atomic package describes a single, logically related and reusable unit of exactly one method element. In a composite package, two or more packages are summarized to become a logical unit. Sub-packages contained in a composite package can be both atomic and composite. Within a composite package, also packages from different method elements can be mixed. The interface of a method fragment again consists of two parts. The fragment specification is a non-formal qualitative description of the purpose and feature overview of the fragment. The adapters of a fragment provide access to the elements contained in the package and facilitate the integration of various method fragments into a method for a specific domain or a specific situation within a domain. Method fragments are stored in a fragment catalog to allow the storage, search and re-use of the fragments. Metamodeling platform components use the fragment catalog and the related method fragments, to integrate and assemble them to create a domain-specific modeling tool.

The continuous use of method fragments in SME and AMME has led to the need to transmit the necessary experiences and needed expert knowledge in an easier way. A possible way to make expert experiences and knowledge explicit and reusable are so called “patterns”. The idea of patterns goes back to the architect Christopher Alexander, who captured and reused design experiences in architecture and civil engineering by using a pattern language [25]. The Enterprise Model Integration approach (EMI) introduced a pattern system for metamodel integration patterns [11, 26, 27]. These patterns help to guide practitioners in integration situations for method fragments related to modeling languages. Metamodels that are primarily complementary employ loose integration patterns. These metamodels are designed to function independently, not relying on the presence of another metamodel. An example would be integrating the BPMN metamodel with the ArchiMate metamodel. They do not rely on each other but could be bi-directionally navigated. For creating a new metamodel or for incorporating elements into an existing one, intermediate integration patterns are utilized. These patterns allow for the partial independence of the new metamodel from the source metamodel, with some elements capable of existing on their own, while others cannot. An example would be mixing elements from a risk & control management metamodel into the BPMN metamodel to allow the modeling of risk and controls as explicit elements in a BPMN process diagram. In contrast, strong integration patterns are applied when creating a new metamodel that fully depends on a source metamodel. In this scenario, the new metamodel cannot exist separately from the source metamodel [26]. An example would be to integrate an ESG-metamodel into the BPMN metamodel to extend the attributes of class “Task” with CO_2 footprint, energy consumption etc. These attributes cannot exist separately of their containing class “Task”.

3 Metamodeling Platforms: Evolutions

In the context of metamodeling platforms various evolutionary steps and directions can be seen. Platform architectures have been extended to use cloud-based services much easier and to allow their interlinkage with the different metamodeling platform components. Combined with mappings and rule based transformation capabilities, powerful integration and interoperability scenarios are supported. Parallel to the usage of cloud-based services, metamodeling platforms itself are now deployed and operated in many cases as cloud-based services. Therefore, deployment and operation features have gained much more importance. And during the recent advent and success of Large Language Models (LLM), the integration of metamodeling platforms and LLM systems moved into focus.

3.1 Metamodeling Platform Architecture

Metamodeling platforms are characterized by a set of logical components and services which can be found in the one or the other way in any kind of metamodeling platform. Figure 4 shows a component architecture of these logical components and is an evolved and extended version of the architecture presented in [1, 9, 11, 28].

The core element of each metamodeling platform is the meta-metamodel providing the metamodeling concepts offered by the platform e.g. model and attribute typing concepts, object-orientation, relationship handling, change-tracking, state-based visualizations, (meta-)model runtime behavior, or (meta-)model extensibility. The metamodel base stores metamodels representing concrete modeling languages. Semantic schemas allow the contextual binding of metamodels to domain-specific semantics and their rule-based validation. Models represented using these modeling languages are stored in the model base. Mechanisms and algorithms used to create, change, use and analyze models are stored in the mechanism base. Procedure models are stored in the procedure base and describe the application of modeling languages and related mechanisms. The instantiation of mechanisms uses a service repository providing internal (pre-defined) services and external services such as cloud-based services. The method fragments catalog is a directory to the method elements stored in the specific bases.

The access and change of method elements (metamodels, procedures, mechanisms etc.) is done via dedicated editors and provided to the end-user via frontend services. These frontend services could be the editors itself e.g. running in a web-browser or specialized user interfaces and UI widgets incorporated into the daily work environment of an end-user such as Atlassian Confluence, Microsoft Teams, Slack etc. The API-based and file-based access services allow the bi-directional, open exchange of related metamodel and model information with other systems [9, 29]. The secure, robust and scalable execution of a metamodeling platform is

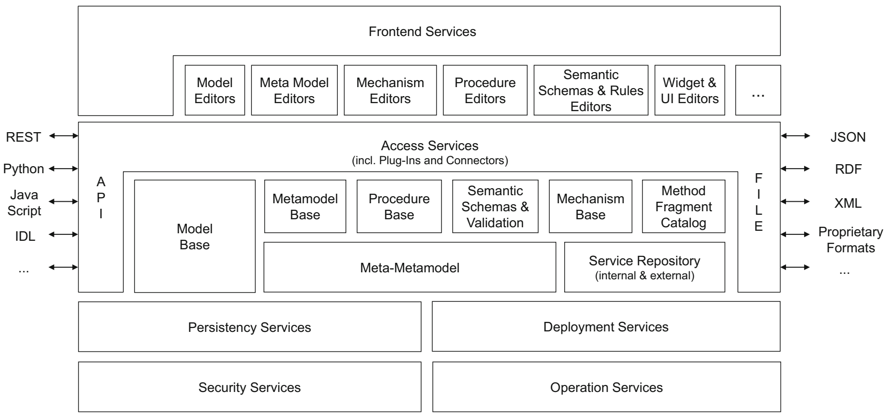


Fig. 4 Component and service architecture of a metamodeling platform

founded on services such as security services (en-/decryption, identity handling, SSO etc.), persistency services (data storage, data retrieval, access protocols etc.), deployment services (containerization, continuous integration/deployment etc.) and operation services (resource allocation, monitoring, scaling etc.).

3.2 Continuous Delivery of Domain-Specific Modeling Tools

Situational usage and agile engineering of modeling methods have lead to lifecycles which a method and its elements go through [10, 13].

Reference [30] provides an extensive overview of approaches to engineer Domain-specific (enterprise) Modeling Methods (DSMM). In addition, the authors systematically embed validation and verification techniques (V&V) into the engineering steps following the five AMME phases. For each phase a detailed proposal is provided which V&V technique should be used, with which quality and goal, which artifacts will be created and which actor is involved.

The iterative, incremental method lifecycle described in [11, 31] consists of the four phases “Acquisition”, “Implementation”, “Introduction” and “Execution”. In the Acquisition phase, objectives and usage goals of the method are defined, followed by requirement identification. This includes initial metamodel sketches and effort estimations. Similar requirements are then consolidated, prioritized, and detailed. The Implementation phase checks for existing method fragments which meet the requirements. These fragments are reused or configured, and new ones are developed, if needed. Integration of these fragments is then documented through various online resources for effective method usage. The Introduction phase involves selecting relevant business units and users, and clarifying technical aspects like infrastructure. Training for scoped users is conducted, with optional piloting for feedback and method fine-tuning. Finally, the Execution phase sees the method rolled out across chosen units and user bases. Support processes are provided alongside, covering methodological, business, and technical aspects. Continuous improvement is driven by data from method usage, and the lifecycle re-starts with the initial phase or ends with the method’s decommissioning when no longer suitable.

Agile engineering approaches such as AMME combined with modern operation environments such as cloud-based environments, containerized/virtualized environments or serverless environments led to additional requirements. Often the DSMM approaches finish with a deployment phase. But using continuous delivery methodologies such as DevOps, operation and monitoring aspects need to be addressed as well. It is expected that upcoming evolutions of DSMM approaches will be extended in this direction. A first proposal is the Continuous Modeling Method Operation (CMMO) approach. It complements AMME with two additional phases (Fig. 5).

In the operation phase particularities of the operation environments need to be considered. This ranges from resource-oriented aspects influencing the performance of the execution of a domain-specific modeling tool, like the availability of main memory, CPU or disk space, the availability of needed third-party service such as data analytics, to information security relevant aspects such as data protection in the used environments. Using operation environments which are priced by load-dependent consumption, also pricing aspects need to be considered during modeling tool operation to avoid any violation of available cost budgets. Storing sensitive data in the modeling tool such as person-related data, confidential product or process data, then also compliance-aspects such as data locations and related legal conditions need to be considered during operation. Using external service providers in operating a distributed ecosystem of modelling services, introduce challenges handling the related uncertainty of external services while providing a reliable and robust solution.

In the monitoring phase the execution behavior of the modeling tool, the underlying operation environment and the usage behavior will be observed. In addition to the possibilities of the underlying metamodeling platform, often additional services are incorporated such as for instance the ELK stack with a popular set of open-source tools consisting of Elasticsearch, Logstash, and Kibana [32]. Each component in the stack serves a specific purpose in data analysis and visualization.

The "Democratization of Modeling" (see Sect. 2.1) led to a broad range of modeling users within an organization or a company. This broad range of users very often needs different maturity levels of tool deployments (stages). Only one type of tool operation environment will not be sufficient in such a context. Tool users in research and innovation oriented business units need the possibility to execute modeling tool experiments or to create tool prototypes for evaluations. To deploy tools quickly without large overhead such as documentation, testing, training etc. The requirements to the underlying operation environment are normally much less

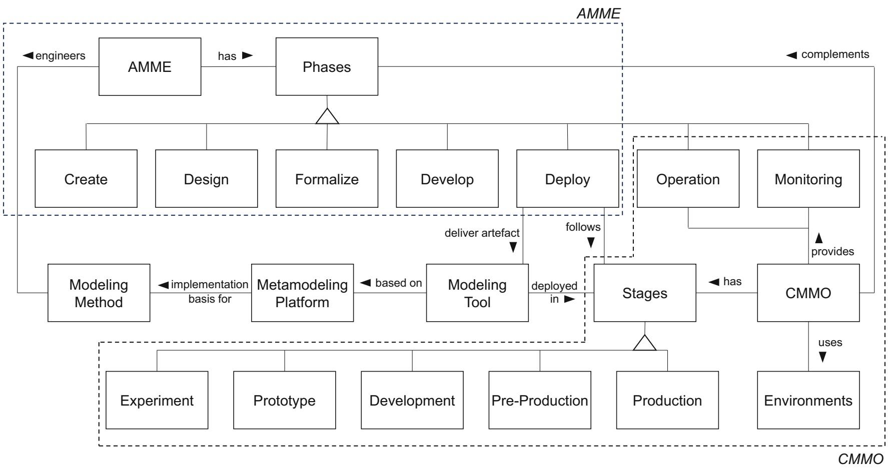


Fig. 5 Continuous Modeling Method Operation (CMMO)

extensive such as in large-scale development, pre-production or production environments. In these environments usually a large volume of developers, meta-modelers and users are working and therefore much higher requirements in performance, availability, backup and data protection are given.

3.3 Integration of Metamodeling Platforms and LLM Systems

Metamodeling Platforms have successfully been used since many years to create and operate domain-specific conceptual modeling tools [16, 17]. Since some years, Large Language Models (LLM) are successfully used for understanding, generating and interacting with human language [33]. They are "large" both in terms of the amount of data they are trained on and their architectural complexity [34, 35]. LLM are a specific subset of Generative AI. Integrating LLM, DSL, and metamodeling platforms can create a powerful synergy and would allow some interesting and useful application scenarios.

Integrating LLMs with DSLs and metamodeling platforms enables the LLM to understand and generate language that is relevant to specific domains, enhancing the accuracy and utility of its responses. The LLM can leverage the structured knowledge from metamodeling platforms, allowing it to provide more informed and contextually appropriate responses, which will lead to enhanced domain-specific understanding and user interaction.

LLMs can assist in automatically generating models and metamodels based on user descriptions or requirements, speeding up both the modeling as well as the metamodeling process. Thanks to the LLM's capabilities, method engineers and domain experts can interact with the metamodeling platform and related modeling tools using natural language, making the interaction more intuitive and accessible. Interesting results from first experiments using ChatGPT with conceptual modeling can be found in [36].

Users can interact with complex modeling tools using natural language, facilitated by the LLM, thereby further democratizing access to advanced modeling capabilities (see also Sect. 2.1). The LLM can translate technical DSL constructs into plain language or domain-independent language constructs to domain-specific concepts, helping bridge the communication gap. An example would be to translate elements of the general purpose EA language ArchiMate to vendor-specific cloud infrastructure mappings for Microsoft Azure, AWS or Google Cloud.

A further application scenario is to provide advanced analytical capabilities. LLMs can analyze models for semantic correctness, consistency, and completeness by understanding the underlying DSL and metamodeling principles. The LLM can provide predictive analytics and intelligent suggestions based on historical data and patterns learned from the metamodeling platform.

LLM system architectures can roughly be represented by three major components. The Model Architecture component defines how the system processes and understands language, the Training and Optimization component ensures that the model learns effectively from its training data, and the Infrastructure and Execution component supports the actual operation and deployment of the model.

LLM systems use embeddings and vector databases [34] for various purposes both in the Model Architecture component as well as in the Infrastructure and Execution component. To represent and understand language, embeddings are used to convert words, phrases, or other textual elements into numerical form, specifically high-dimensional vectors. These embeddings capture the semantic meanings and relationships of the language elements, which are essential for the model to understand and process language. Modern LLMs often use context-aware embeddings (like those generated by Transformer models) that represent words in the context of their surrounding text, leading to a more nuanced understanding of language. They enable semantic search capabilities, allowing LLMs to find information based on the meaning and context of the query, rather than relying solely on keyword matching.

Vector databases are specialized systems designed to store and manage high-dimensional vector data like embeddings. They are optimized for operations like similarity search, which is fundamental in tasks such as retrieving relevant information. These databases allow LLM systems to efficiently handle large volumes of embeddings, ensuring scalability and performance, especially in applications involving real-time processing or large datasets.

In cases where LLMs are integrated with external domain-specific knowledge such as from metamodeling platforms, embeddings and vector databases can be used to incorporate and leverage this external domain-specific model data, enhancing the LLM's expertise in that particular area. The proposed integration approach is to feed the different bases of the metamodeling platform, e.g. the metamodel base, the model base, the procedure base or the base of semantic schemas and validation rules, into the embeddings layer generating for each base one or more vector databases (see Fig. 6). During the generation step, for each base one or more natural language representations are generated which are then processed by the LLM system to feed the embeddings and the related vector databases.

4 Outlook

Looking towards the future of metamodeling platforms, it is evident that several key areas will shape their evolution and utility. This outlook aims to highlight four important aspects.

Integration with emerging technologies: a significant trend in the advancement of metamodeling platforms is their integration with cutting-edge technologies. The synergy between metamodeling and advanced artificial intelligence, including large language models, opens up new possibilities for automated model generation, predictive analytics, and enhanced decision-making capabilities. Moreover, the integration with domain-specific cloud services promises better contextualization for specific problems under consideration. Easier integration and usage of cloud

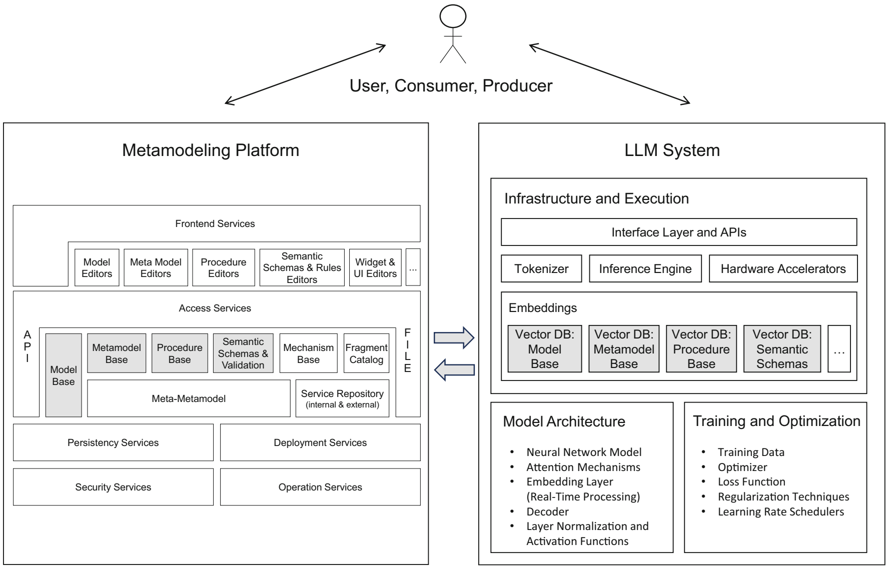


Fig. 6 Integration of metamodeling platforms and LLM systems

computing technologies will improve scalability expanding the potential of metamodeling platforms in handling complex and large-scale modeling tasks.

Democratization and accessibility: the future of metamodeling is also strongly tied to its democratization and increased accessibility. This is where the concept of self-service as well as the no code/low code movement comes into play. This approach empowers users like citizen-developers to create and modify models and even their metamodels without extensive coding knowledge, essentially enabling self-service in model development. It aligns perfectly with the need for domain-specific solutions, as users can tailor models to their specific industry requirements without deep technical expertise.

Runtime metamodel adaptation: one of the key challenges is the need for runtime metamodel adaptation. As business environments and requirements change rapidly, metamodeling platforms must be capable of adapting in real-time and to provide easy-to-use staging mechanisms. This requires not only robust underlying technology but also a user-friendly approach and data security.

Open innovation: it is expected that domain-specific aspects will remain a very important driver for the future of metamodeling platforms. Therefore, targeted research and innovation approaches are needed. Collaborative efforts between academia, industry, and users will be crucial in driving this applied research, guaranteeing that metamodeling platforms evolve to meet the related requirements.

References

1. Karagiannis, D., Kühn, H.: Metamodelling platforms. Invited paper. In: Bauknecht, K., Tjoa, A.M., Quirchmayer, G. (eds.) Proceedings of the Third International Conference ECWeb 2002 - Dexa 2002, Aix-en-Provence, France, September 2–6, 2002, LNCS 2455, p. 182. Springer, Heidelberg (2002)
2. Tolvanen, J.-P., Rossi, M.: MetaEdit+: defining and using domain-specific modeling languages and code generators. In: OOPSLA'03 Companion of the 18th Annual ACM SIGPLAN Conference, pp. 92–93. ACM (2003)
3. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework, 2nd edn. Addison-Wesley Professional (16 December 2008)
4. Jeusfeld, M.A., Jarke, M, Mylopoulos, J.: Metamodeling for Method Engineering, p. 424. The MIT Press, Cambridge. ISBN 978-0-262-10108-0. http://conceptbase.sourceforge.net/2021_Metamodeling_for_Method_Engineering.pdf (2009). Accessed 28 Jan 2024
5. ADOxx Metamodeling Platform: https://www.adoxx.org/. Accessed 28 Jan 2024
6. MetaEdit+ Products: https://www.metacase.com/products.html. Accessed 28 Jan 2024
7. EMF Software: https://eclipse.dev/modeling/emf/. Accessed 28 Jan 2024
8. Fill, H.-G., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. Enterp. Model. Inf. Syst. Archit. **8**(1), 4–25 (March 2013)
9. Kühn, H., Murzek, M.: Interoperability issues in metamodelling platforms. In: Proceedings of the 1st International Conference on Interoperability of Enterprise Software and Applications (I-ESA'05), Geneva, Switzerland, pp. 215–226. Springer (February 2005)
10. Henderson-Sellers, B., Ralyté, J., Brinkkemper, S. (eds.): Situational Method Engineering: Fundamentals and Experiences. Proceedings of the IFIP WG 8.1 Working Conference, 12–14 September 2007. Springer, New York (2008)
11. Kühn, H.: Methodenintegration im Business Engineering (English: Method Integration in Business Engineering). PhD Thesis, University of Vienna (April 2004)
12. Faisal, T.: Blog Post "Sustainability and enterprise architecture: how do they fit together?". https://www.boc-group.com/en/blog/ea/sustainability-and-ea/. Accessed 28 Jan 2024
13. Karagiannis, D.: Agile modeling method engineering. In: PCI'15 - Proceedings of the 19th Panhellenic Conference on Informatics, October 2015, pp. 5–10. ACM Digital Library. https://doi.org/10.1145/2801948.2802040. Accessed 28 Jan 2024
14. The OMiLAB Community: Development of conceptual models and realization of modelling tools within the ADOxx meta-modelling environment: a living paper. In: [15]. https://link.springer.com/chapter/10.1007/978-3-030-93547-4_2 (2022)
15. ADOxx-based Tools from the OMiLAB Community: https://www.omilab.org/activities/projects/. Accessed 28 Jan 2024
16. Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.): Domain-Specific Conceptual Modeling: Concepts, Methods and Tools. Springer (2016)
17. Karagiannis, D., Moonkun, L., Hinkelmann, K., Utz, W. (eds.): Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools. Springer (2022)
18. ADONIS and ADOIT Community Editions: https://www.adonis-community.com/en/ and https://www.adoit-community.com/en/. Accessed 28 Jan 2024
19. BOC Management Office Product Suite: ADONIS, ADOIT and ADOGRC. https://www.boc-group.com/en/adonis/, https://www.boc-group.com/en/adoit/ and https://www.boc-group.com/en/adogrc/. Accessed 28 Jan 2024
20. Sandkuhl, K., Fill, H.-G., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A., Schwabe, G., Uludag, Ö., Winter, R.: From expert discipline to common practice: a vision and research agenda for extending the reach of enterprise modeling. Bus. Inf. Syst. Eng. **60**(1), 69–80 (February 2018)

21. Gaughan, D., Natis, Y., Alvarez, G., O'Neill, M.: Future of Applications: delivering the composable enterprise. Gartner Research Report, G00465932, February 2020. https://www.gartner.com/en/doc/465932-future-of-applications-delivering-the-composable-enterprise. Accessed 28 Jan 2024
22. Henderson-Sellers, B., Gonzalez-Perez, C., Ralyté, J.: Comparison of method chunks and method fragments for situational method engineering. In: Proceedings of the 19th Australian Conference on Software Engineering (ASWEC 2008), Perth, Australia, pp. 479–488. IEEE (2008)
23. Ralyté, J., Jeusfeld, M.A., Backlund, P., Kühn, H., Arni-Bloch, N.: A knowledge-based approach to manage information systems interoperability. Inf. Syst. Int. J. **33**(7–8), 754–784 (November–December 2008)
24. Ralyté, J., Backlund, P., Kühn, H., Jeusfeld, M.: Method chunks for interoperability. In: Embley, D.W., Olivé, A., Ram, S. (eds.) Proceedings of the 25th International Conference on Conceptual Modelling (ER'2006), Tucson, Arizona, USA, November 6–9, 2006, LNCS 4215, pp. 339–353. Springer
25. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press (1977)
26. Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise model integration. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) Proceedings of the 4th International Conference EC-Web 2003 - Dexa 2003, Prague, Czech Republic, September 2003, LNCS 2738, pp. 379–392. Springer
27. Zivkovic, S., Kühn, H., Karagiannis, D.: Facilitate modelling using method integration: an approach using mappings and integration rules. In: Österle, H., Schelp, J., Winter, R. (eds.) Proceedings of the 15th European Conference on Information Systems (ECIS2007) – "Relevant Rigour - Rigorous Relevance", pp. 2038–2050. ECIS, St. Gallen, Switzerland (June 2007)
28. Zivkovic, S., Kühn, H., Murzek, M.: An architecture of ontology-aware metamodelling platforms for advanced enterprise repositories. In: Camp, O., Hammoudi, S. (eds.) Proceedings of the 1st International Workshop on Advanced Enterprise Repositories (AER 2009), Co-located with 11th International Conference on Enterprise Information Systems (ICEIS 2009), Milano, Italy, pp. 95–104. ICEIS (6 May 2009)
29. Riggio, R., Ursino, D., Kühn, H., Karagiannis, D.: Interoperability in meta-environments: an XMI-based approach. In: Pastor, O., Falcão e Cunha, J. (eds.) Proceedings of the 17th International Conference on Advanced Information Systems Engineering (CAiSE'05), Porto, Portugal LNCS 3520, pp. 77–89. Springer (June 2005)
30. Ma, Q., Kaczmarek-Heß, M., de Kinderen, S.: Validation and verification in domain-specific modeling method engineering: an integrated life-cycle view. Softw. Syst. Model. **22**(2), 647–666 (April 2023)
31. Kühn, H., Murzek, M., Specht, G., Zivkovic, S.: Chapter 7 "Model-driven development of interoperable, inter-organisational business processes". In: Charalabidis, Y. (ed.) Interoperability in the Digital Public Services and Administration: Bridging E-Government and E-Business, pp. 119–143. ISBN 978-1-61520-887-6, IGI Global (2010)
32. The ELK: Stack – Elasticsearch, Logstash, Kibana. https://www.elastic.co/de/elastic-stack. Accessed 28 Jan 2024
33. Wikipedia. Large Language Model: Definition, explanation and history. https://en.wikipedia.org/wiki/Large_language_model. Accessed 28 Jan 2024
34. Bornstein, M., Radovanovic, R.: Emerging architectures for LLM applications. https://a16z.com/emerging-architectures-for-llm-applications/. Accessed 28 Jan 2024
35. Choi, N.: The architecture of today's LLM applications. In: Github Blog. https://github.blog/2023-10-30-the-architecture-of-todays-llm-applications/. Accessed 28 Jan 2024
36. Fill, H.-G., Fettke, P., Köpke, J.: Conceptual modeling and large language models: impressions from first experiments with ChatGPT. Enterp. Model. Inf. Syst. Archit. **18**(3) (2023)

A Metamodel-Driven Architecture for a Unified Approach to Governance, Risk, Compliance and Performance

Christian Lichka

Abstract Regulatory requirements—often summarized under the term compliance—are crucial to every company. Recent years have shown that these demands are continuously evolving. Their general conditions—especially with focus to risks—are susceptible to continuous and at times accelerating changes influenced by local, national, and global factors. This paper derives three findings to improve compliance management by addressing a unified approach that extends governance, risk, and compliance (GRC) with a performance dimension and proposes a metamodel-driven unified architecture blueprint.

- Finding #1: The advanced degree of digitalization in companies enforces the handling of compliance requirements to be executed by and with the help of adequate software programs to master the amount, quality, complexity, and traceability of those requirements.
- Finding #2: A modern approach to deal with compliance is switching from a pure cost approach of fulfilling requirements to an approach and tooling that delivers direct financial added value and differentiation in the market. Compliance becomes a relevant component of corporate performance.
- Finding #3: Compliance IT implementation must be made more flexible and generalized in such a way that the topics outlined in the aforementioned findings are covered. This requires a cross-domain unified data and software architecture that is explicitly setup for Governance, Risk, Compliance, and Performance (GRCP).

Keywords Governance · Risk management · Compliance · Company performance · Metamodeling · Regulatory requirements · GRC software

C. Lichka (✉)
BOC Group, Vienna, Austria
e-mail: christian.lichka@boc-group.com

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_8

1 Evolvement of Compliance Requirements

Over the past years, the landscape of compliance standards has evolved significantly, marked by the introduction and refinement of various frameworks and regulations worldwide. These standards collectively reflect a global trend towards more rigorous, transparent, and accountable regulatory environments, aiming to enhance various aspects such as data protection, financial stability, equal treatment, consumer safety, environmental aspects, or information security (e.g. in [1–4]). The following list contains an overview of selected, prominent guidelines of the last decades:

1. Basel I (1988): The Basel Committee on Banking Supervision introduced Basel I to establish minimum capital requirements for banks. This was the first attempt to set global banking standards following financial disruptions in the 1980s [5].
2. Basel II (2004): Basel II expanded on the original framework by introducing more sophisticated risk and capital management requirements. It focused on three pillars: minimum capital requirements, supervisory review, and market discipline [6].
3. ISO 9001 (First published in 1987, with revisions, latest in 2015): This standard set criteria for a quality management system. It is based on quality management principles including a strong customer focus, the involvement of high-level company management, a process approach, and continuous improvement [7].
4. MaRisk (2005 in Germany): The Minimum Requirements for Risk Management (MaRisk) were introduced by the German Federal Financial Supervisory Authority (BaFin) as a comprehensive framework for risk management in banks [8].
5. ISO 27000 series (First published in 2005, latest revision in 2022): Known as ISO 27 K, these standards provide a model for establishing, implementing, operating, monitoring, and improving an Information Security Management System [9, 10].
6. ISO 31000 (First published in 2009, latest revision in 2018): This standard provides guidelines on risk management. It seeks to provide principles, a framework, and a process for managing risk. It can be used by any organization regardless of its size, activity, or sector [11].
7. ISO 19600 (2014): ISO 19600 provides guidelines for establishing, developing, implementing, evaluating, maintaining, and improving an effective and responsive compliance management system within an organization. This standard has been renewed and replaced by ISO 37301:2021 [12, 13].
8. Basel III (2010-2011, phased implementation till 2019): In response to the financial crisis of 2007-2008, Basel III introduced more stringent capital requirements, new regulatory requirements on bank liquidity, and leverage [14].
9. GDPR (General Data Protection Regulation, or DSGVO in German, 2018): A significant shift in data privacy, GDPR gave individuals in the EU greater control over their personal data and imposed strict data processing guidelines on organizations [15].

10. CSRD (Corporate Sustainability Reporting Directive, proposed in 2021): The CSRD is an initiative by the European Union to extend the sustainability reporting requirements to all large companies and all companies listed on regulated markets (except listed micro-enterprises) [16].
11. ESG (Environmental, Social, and Governance criteria): While not a single standard, ESG criteria have become increasingly important in business and investment decision-making. Over recent years, a variety of standards and frameworks have been developed to guide companies in ESG reporting and performance [16–18].

Having named standards with national and international focus, the above-mentioned list includes a stronger focus on EU countries. However, this list could be massively enriched when extending the focus of observation on other regions. The United States for example has introduced several key compliance standards to address various aspects of corporate governance, financial practices, and privacy concerns in the last decades. The Sarbanes-Oxley Act (SOX) of 2002 established rigorous oversight of corporate financial reporting to prevent fraud and protect investors [19] and extended the organization business and process management with enterprise risk domains [20]. The Dodd-Frank Wall Street Reform and Consumer Protection Act of 2010, enacted in response to the 2008 financial crisis, significantly reformed financial regulation and aimed to reduce systemic risks [21]. The Health Insurance Portability and Accountability Act (HIPAA) of 1996 safeguards sensitive patient health information, setting standards for privacy and security in the healthcare sector [22]. These standards as just a few examples collectively enhance corporate accountability, financial integrity, and data security in the U.S. business landscape.

In the evolving landscape of regulatory compliance, a significant trend is emerging beyond the isolated implementation of individual standards. Organizations are increasingly recognizing the necessity of developing integrated corporate systems that holistically address a myriad of compliance requirements. This shift is driven by the growing complexity and interrelatedness of compliance-related topics, such as risk management, IT- and data security. Traditional approaches, which often involved siloed strategies for each regulation, are becoming inadequate in the face of this complexity. Instead, there is a move towards more cohesive, overarching frameworks that can encompass and harmonize the various aspects of compliance [22].

A specific challenge is the multifaceted nature of risks and scenarios addressed in compliance regulations. Modern business operations are intrinsically complex and interconnected, making it challenging to tackle issues like operational risks, cybersecurity threats, data privacy concerns, and financial regulations independently. Each of these areas can influence and overlap with others, necessitating a comprehensive approach. For instance, a data breach in one department can have far-reaching implications, affecting compliance with data protection regulations, financial reporting standards, and even customer trust. By adopting an integrated system, organizations can ensure that their response to one area of compliance does not inadvertently create vulnerabilities or non-compliance in another.

This holistic approach is also being reflected in newer regulatory frameworks. A notable example is the Swiss Financial Market Supervisory Authority's (FINMA) "Rundschreiben für operationelle Risiken und Resilienz" (Circular on Operational Risks and Resilience) [22]. This regulation underscores the importance of having structured, overarching risk management systems that are not just confined to specific areas but are capable of addressing the full spectrum of operational risks and building organizational resilience. Such regulatory norms are pushing companies towards a more systemic approach to compliance, where the focus is not just on meeting the minimum requirements of individual standards but on creating a robust, comprehensive framework that can adapt to various compliance challenges in a unified manner.

It is obvious that implementing a systemic approach to compliance in today's complex regulatory environment necessitates the use of IT software [23]. Technology is essential due to its ability to handle large volumes of data, automate compliance processes, and provide real-time monitoring and reporting. IT systems are particularly adept at identifying and managing interconnected risks across various compliance areas, a task too complex for manual approaches. Additionally, software solutions ensure up-to-date adherence to changing regulations and help in maintaining comprehensive audit trails. The integration of domain specific functionality and metamodeling capabilities extends software in semantically enriched data structures and scenario flexibility. In summary, the use of IT software is not just beneficial but critical for organizations aiming to achieve an integrated and efficient compliance framework.

2 From Compliance to Company Performance

The above-mentioned aspects strengthen the ongoing shift from viewing compliance solely as a regulatory obligation to recognizing it as a driver of performance and financial success. This evolution is grounded in the fact that effective compliance management, especially in areas like ESG (Environmental, Social, and Governance), Risk Management, and quality standards such as ISO 9001, aligns closely with operational excellence and market differentiation. Companies adept in these compliance areas not only mitigate risks and avoid penalties but also gain efficiencies and earn stakeholder trust. This strategic integration of compliance with business performance offers a distinct competitive edge. Adherence to ESG criteria, for example, underscores a commitment to sustainable and ethical practices, resonating with the values of a growing demographic of conscious consumers and investors. Similarly, meeting stringent financial or quality regulations signals reliability and trustworthiness, key factors in attracting premium customers and fostering loyalty. Thus, effective compliance management transcends its traditional role, emerging as a pivotal element in shaping a company's market identity and driving both financial performance and differentiation in an increasingly value-driven business landscape.

Significant influence on company performance can be seen as an impact on key financial metrics like profit margins, ROI, and EBITDA [25]. Effective compliance reduces legal penalties and operational disruptions, directly safeguarding profit margins. Let's take a look at one exemplary company KPI - Profit Margin:

$$\text{Profit Margin } = \text{NetIncome}/\text{Revenue}$$

whereas

$$\text{Net Income} = \text{Total Revenue} - \text{Total Expenses}$$

Compliance management will improve a company's profit margin by influencing both components of the profit margin formula: Net Income and Revenue.

2.1 Increasing Net Income:

1. Reducing Expenses: Effective compliance helps in avoiding the costs associated with non-compliance, such as fines, legal fees, and penalties. By minimizing these expenses, net income is improved. For instance, a company adhering to environmental regulations avoids fines for violations, thereby reducing its total expenses.
2. Preventing Operational Disruptions: Compliance with regulations should lead to more efficient and streamlined operations. This can prevent costly disruptions, like production halts due to safety violations or data breaches. Efficient operations contribute to a reduction in operational expenses, thus improving net income.
3. Enhancing Reputation: Companies with strong compliance records often enjoy a better reputation, which should lead to increased customer trust and loyalty, potentially translating to higher sales and customer retention.

2.2 Increasing Revenue:

1. Market Differentiation: Compliance, especially in areas like ESG or QM, can be a market differentiator, attracting customers who prioritize such criteria. This should lead to increased visibility, sales, and consequently, higher total revenue.
2. Access to New Markets: Compliance with international standards (like ISO certifications or regional compliance certificates) can open up new markets, expanding the customer base and increasing revenue.

The same can be derived for other company or investor-relevant KPIs such as ROI, EBITDA, Debt-to-Equity, or even Customer Retention.

Summarized, a modern approach to dealing with compliance is switching from a pure cost approach of fulfilling the requirements to an approach and tooling that delivers direct financial added value and differentiation in the market. Compliance becomes a relevant component of corporate performance.

3 A Cross-Domain Data Model for Governance, Risk, Compliance and Performance

In light of the multifaceted and interrelated nature of compliance and its impact on company performance, it is evident that a flexible and interconnected data model for software applications is essential. The data models need to be semantically enriched. Metamodeling concepts [26–28] will be applied to ensure that the data model accommodates the complexities and interdependencies of various compliance areas at once. This will allow for a comprehensive understanding and representation of how governance, risk, compliance, and performance are covered and interlinked within the organization.

Agile modeling plays a crucial role in the development of such data models that can cover holistic enterprise aspects. Referring to "Agile Modeling Method Engineering [29]" agility in enterprise modeling encompasses the ability to quickly adapt the models to changing business requirements and incorporate feedback from stakeholders effectively. This agility is vital in the context of compliance and performance management, where regulatory changes and market dynamics necessitate fast and adaptive responses.

The challenge arises from covering not only dynamic data models due to the fact of evolving and continuously changing compliance requirements. The need for cross-domain (i.e. cross-regulatory requirements) semantic enrichment of the data by addressing an overarching terminology (e.g. risk understanding) has to be covered as well. Finally, provided functionality must be generic on the one side (fitting for multiple standards) but still pre-defined and highly configurable to fit individual standards (e.g. risk assessments) and solutions [38].

Reflecting the aspect of cross-domain semantics first—which is one of the most challenging ones. Compliance refers to the effort of an organization to adhere to relevant laws, regulations, and standards while also minimizing the risk of legal or financial penalties. Where P. Vicente and M.M. Da Silva [30] give a good overview on the complexity of risk and compliance terms in their conceptual model, most of the focus goes in the direction of procedure models. Challenges arise when overarching compliance and risk concepts are needed, such as those requested in 2022 by Finma local authorities in Switzerland [24]. These regulatory statements are exemplary for upcoming over-arching requirements, which therefore need a cross-domain data and application model to deal with.

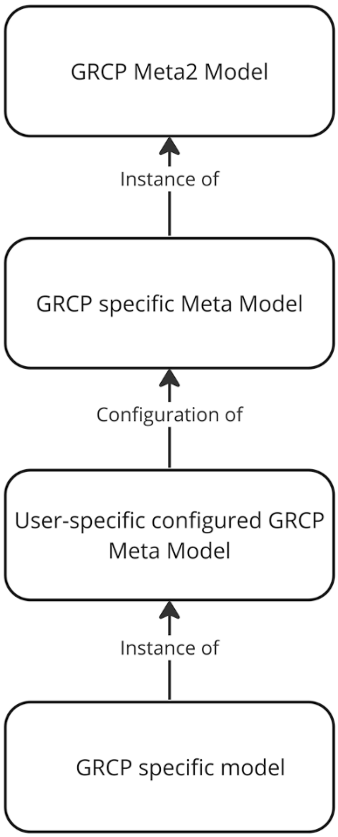
Such a cross-domain model has been derived and will be presented in the following. It shall be understood as a Meta-Metamodel (Meta^2 -model) according to Visic et al. [31] to cover any kind of GRC-related metamodel. On that basis compliance-standard specific data- and functionality methods (Modeling Method Engineering) can be derived. It will be called Governance, Risk, Compliance & Performance meta^2 model (GRCP) to make the cross-domain and impact scope explicit.

4 A Governance, Risk, Compliance & Performance (GRCP) Meta^2 Model

Deriving the GRCP Meta^2 -Model is based on the findings of [26, 28] on how model and modeling hierarchies are defined. The following graphic shows the corresponding hierarchy for the GRCP approach (Fig. 1).

The Meta^2 -Model is a generalization based on the various instances of meta-models needed (GRCP-specific Meta-Models) to cover the different compliance

Fig. 1 GRCP meta model hierarchy



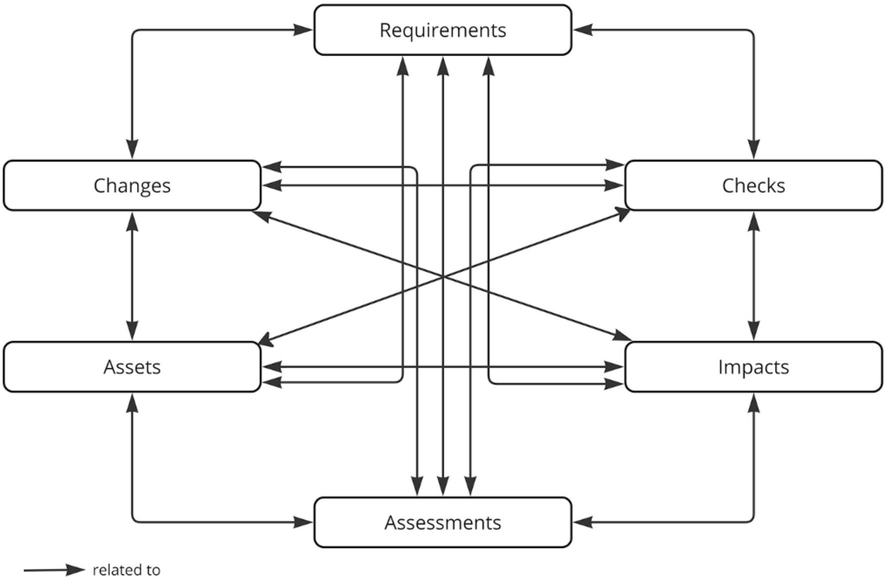


Fig. 2 GRCP cross domain meta^2 model

requirements and standards. Practice has shown, that customer-specific adaptations are always needed on top, therefore resulting in a user-specific configuration of a GRCP Meta-Model [32]. It has to be noted, that based on the fact, that companies are always affected by a variety of compliance requirements, multiple GRCP-specific meta-models (i.e. methods) are applied at the same time (within one or more software solutions). Therefore it is obvious, that a generalization in a Meta^2 GRCP model offers semantic and functional advantages, which will be dealt with later (Fig. 2).

The GRCP cross-domain Meta^2 -Model contains Meta^2 classes and their relations. Meta^2 classes are a generalization of classes (data objects) needed to fulfill the various compliance standards (i.e. of classes used in GRCP-specific meta-models). Table 1 lists further details of the Meta^2 classes.

To take advantage of a Meta^2 level, concepts must be drawn to the Meta^2 level via generalization. Semantics, data storage, and functional concepts are particularly relevant. By doing this, these concepts are later inherited by instantiation on any GRCP meta-models (i.e. GRCP methods) derived.

Table 1 Meta^2 class table

Meta^2 class	Meaning	Derived meta-classes (Proposal)	Examples	Inherited functionality (selected)
Requirements	All types of conditions to be achieved, whether of a voluntary or regulatory nature	• Generic Control objectives • Targets • Objectives • Requirements • Standard Operating procedures • Rules • All above mentioned in a domain specific way	• Regulatory requirements • ISO standards • Domain specific such as IT Sec SOPs, Financial Control Objectives, . . .	• Detailing descriptions • Attaching documents (such as legal documents) • Versioning • Validity status • Multiple assessment relations • Ownership • Classification
Checks	Any kind of validating activity with the focus on comparing a status with an expectation	• Control activity • Control execution • Control test • Audit activity • Task	• Documented control activity within a process/SOP • Financial audit tasks • Certification test	• Target status description • Attaching and typing of documents such as findings, execution reports • Relations to requirements (derived from) • Versioning • Status • Relations to follow up classes such as Impacts, Changes • Ownership

(continued)

Table 1 (continued)

$Meta^2$ class	Meaning	Derived meta-classes (Proposal)	Examples	Inherited functionality (selected)
Impacts	Any kind of event, whether real or not, that has an effect on the subject	• Risk • Opportunity • Scenario • Loss/Near loss • All above mentioned in a domain specific way	• Compliance risk • Strategic risk • Business opportunity	• Versioning • Validity status • Multiple assessment relations • Ownership
Assessments	Any kind of measurement or evaluation	• Risk assessment • Control assessment • Audit assessment • Survey • Estimations	Domain-specific assessment classes such as financial risk assessment, strategic risk assessment, compliance assessments, Design & Effectiveness Assessments	• Qualitative and quantitative single and multi-dimension assessment • Versioning • Validity status • Aggregation mechanisms
Assets	Any kind of generic element (category) that a company has or can have at its disposal to achieve its goals and that can be relevant for compliance	• Roles • Functions • Processes • IT applications • Capabilities • Resources • Employees • Locations • SOPs	• All financial processes that are subject to financial/operational risks • All SOPs that are subject for compliance according to a specific regulation	• Assets documentation • Ownership • Versioning • Display of all related GRC contents ($meta^2$ relations)

5 Reflections on a GRCP Meta^2-based Application

The advantages of meta-modeling shall not only be applied to data flexibility and semantics but also to functional aspects of a GRCP application. This paper argues for three aspects to be covered:

- Semantic enrichment done on the Meta^2 model to be inherited on concrete compliance method implementations
- A corresponding data storage capability that covers semantic requirements
- Functional concepts being implemented and assigned on Meta^2 level (Fig. 3)

Core of the implementation proposal are constructs around the class element. (Meta^2/Meta)-Classes have attributes to specialize for GRCP-specific Meta-Models and to allow user-specific configuration. Data storage and data representation are explicitly split to allow flexibility in data storage and context-specific data representation, independent from data storage. Based on concepts from business intelligence [33] dealing with multidimensionality is essential. This is especially true in GRCP as scenarios from performance management are covered as well as well as muti-dimensional compliance. Compliance structures and company structures are base dimensions in GRCP. Dimensionality [33] always goes hand-in-hand with aggregation functionality. Flexibility in GRCP is ensured by defining dimensions as well via relations (e.g. between risks or to risk groups) as well via attributes (e.g. time-aspects defined via attributes). As dimensionality already

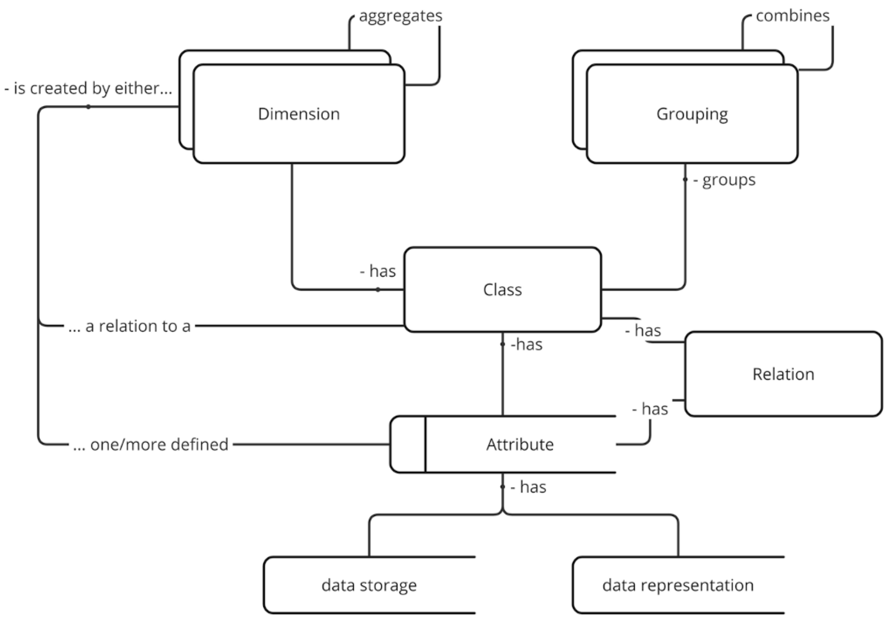


Fig. 3 Core of the Meta^2 Model around the Class Object

provides structuring, grouping is the second concept to structure data without the need of dimensionality.

By applying the concepts shown in the graphic above on the cross-domain Meta^2 GRCP model, a meta-model based application can be setup to cover the data structure needs of GRCP. Semantic enrichment is done via inheritance of semantic information already applied on Meta^2 level and extended by the definition of GRCP-specific meta-models.

Leveraging meta-modeling also implies assigning functionality on a more generalized meta-level. By doing this, derived meta-models are directly equipped with context-specific functionality. GRCP-specific functionality can therefore be assigned at Meta^2 level (see table above) and inherited by instantiating GRCP-specific meta-models (Fig. 4).

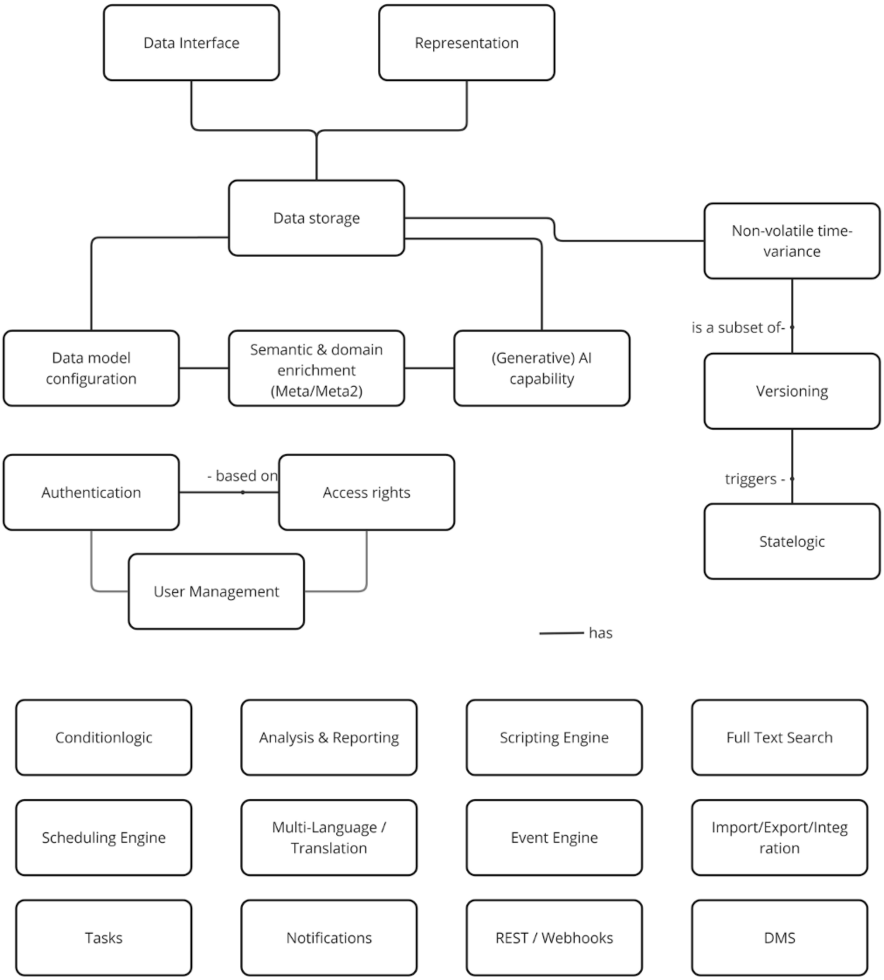


Fig. 4 GRCP functional concepts

The GRCP functional concepts shown in the graphic above are to be understood as a capability catalogue with selected, important relations highlighted [34]. These concepts should already by assigned on Meta^2 level. Core element is suitable data storage, as GRCP invokes various requirements to data storage due to versioning, historicization, and audit trails (state logic, versioning, non-volatile time-variance). These requirements are extended by current application functionality expectations such as (generative) AI or flexibility in the meaning of composable architectures [35].

Further capabilities defined are from state-of-the-art business applications such as authentication based on access and user management, condition logic or scripting.

In conclusion, by leveraging the Meta^2 model, companies can inherit semantic enrichment and flexibility, data storage capabilities, and functional concepts in their specific compliance method implementations. The Meta^2 model allows for the inclusion of multidimensional aspects, such as performance management and cross-domain compliance and allows for the direct assignment of context-specific functionality at Meta^2 level. This approach enables the setup of a meta-model based software application that covers the diverse data structure needs of GRCP by instantiation of multi-scenario meta-models.

6 A Corresponding Procedure Model

To ease anchoring in a company, a suitable process in the sense of a procedure model is presented, which is already being used successfully with the metamodel-based GRC application ADOGRC [36].

The procedure model is subject to the same cross-domain requirements, making it a recommended approach for integrating all applicable compliance requirements within an organization (Fig. 5).

Six sequential stages characterize the procedure model, which is typically anchored in an annual cycle. Periodicity is mainly defined by each compliance topic individually, whereas business/calendar years and re−/certification schedules are the most common patterns. However recent years have shown an increased demand for short-term patterns in compliance management to allow more agile and rapid reactions to changed circumstances. In the latter case, selected steps are performed on ad-hoc schedules, e.g. execute new assessments, perform changes on implementations, or analyze situational changes. Contents of the stages are explained in the following sections.

6.1 Stage Scope

Scoping refers to the crucial step of identifying all relevant compliance components (typically risks) that apply to an organization. Some frameworks, e.g. ISO 27000 [10] or ONR 49000 [37] already provide (risk) lists that can serve as starting point

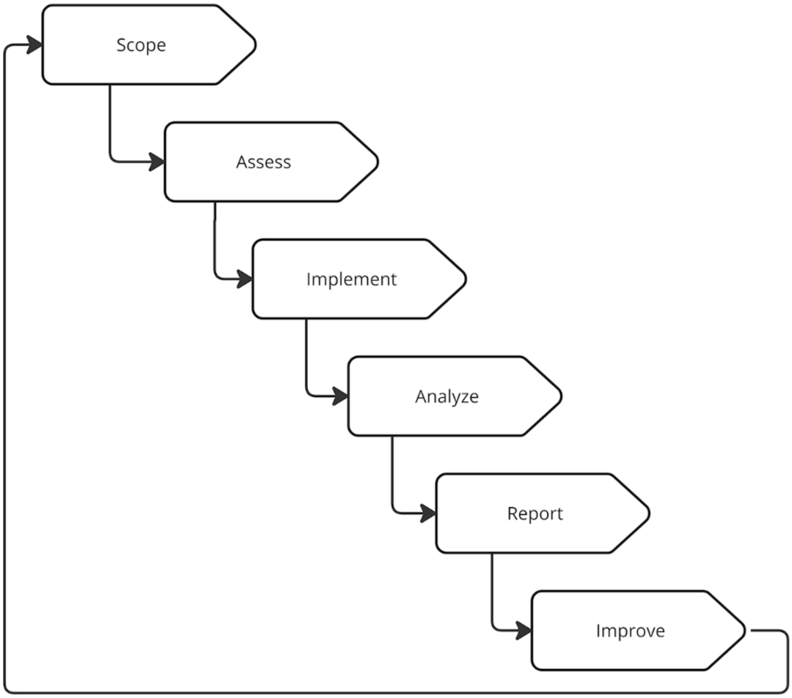


Fig. 5 GRCP procedure model

for scoping activities. Software applications can support this step by providing guidance, reference content, and transparent audit trails on what and why something has been scoped or descoped.

6.2 Stage Assess

Since GRCP typically deals with a large number of topics, prioritization is essential. Being able to prioritize means having a resilient basic understanding of the most relevant impact, urgency, and status of compliance issues. To achieve this, the scoped topics are assessed, e.g. through scenario or risk assessments, control tests, compliance fits etc.

6.3 Stage Implement

Implementation includes all activities to anchor the regular and defined handling of compliance requirements in the company. This includes, for example, adapted processes, new control steps, computer-aided or automated validations and clear compliance-related responsibilities set up.

6.4 Stage Analyze

To anchor compliance, and in particular the handling of risks and controls in the company in such a way that performance of the company is positively influenced, ongoing analysis is necessary. This concerns the dimensions of changing, or the need to expand or reduce the compliance portfolio. In addition, this step includes an in-depth examination of the functioning and fit (design and effectiveness) of the previously defined measures.

6.5 Stage Report

A high percentage of compliance topics are subject to some form of external audit/oversight. It is a common best practice to establish in addition an internal structure through internal audits that takes care of compliance checks. For such defined target groups, reporting is a crucial step to be informed about the status, to recognize changes, and finally to prove regulatory fit.

6.6 Stage Improve

This step is essentially taken from CIP concepts [7]. It aims to continuously question (analyze) compliance processes per se as well as the specific compliance-related contents and to improve them based on these findings. From a performance perspective in particular, this step is essential to ensure a healthy cost-benefit ratio of compliance implementation.

7 Conclusion and Ideas for Future Research

The central work result of this paper is the cross-domain Meta^2 model for GRCP. It allows the instantiation of various GRCP metamodels to cover regional, national, and international compliance requirements. Extending the Meta^2 model by connecting functional concepts provides the basic architecture for a Meta^2 model-based software application for GRCP. To be able to understand how to proceed and how to anchor compliance in a company a procedure model has been presented.

As an input for future research the contrast between the constantly growing volume and complexity of compliance requirements and the even greater dynamism of market changes, (threat) scenarios, and findings shall be highlighted. In other words, the agility and dynamism of the reality outside of companies contrasts with an ever-increasing number of structural requirements within companies. This requires capabilities to respond to compliance scenarios in a way that is highly structured in terms of both professional expertise and IT technology, while also being highly flexible and dynamic.

References

1. BITKOM: Leitfaden Compliance - Rechtliche Anforderungen an ITK-Unternehmen, vol. 2019
2. Austrian Standards. austrian-standards.at (Accessed: 8 Jan. 2024)
3. ISO Standards. iso.org (Accessed: 2 Jan. 2024)
4. Montenegro-Almonte, A., Sultan, A., Ren, A.: US Compliance Requirements. https://globalinvestigationsreview.com/guide/the-guide-compliance/first-edition/article/us-compliance-requirements (Accessed: 4 Jan. 2024)
5. Basle Committee On Banking Supervision: International Convergence of Capital Measurement and Capital Standards, July 1988
6. Basle Committee On Banking Supervision: Internationale Konvergenz der Eigenkapitalmessung und der Eigenkapitalanforderungen II, June 2004
7. ISO 9000 family - quality management. iso.org (Accessed: 8 Jan. 2024)
8. MaRisk in der Fassung vom 20.12.2005 – Dokumentensammlung. https://www.bundesbank.de/de/aufgaben/bankenaufsicht/einzelaspekte/risikomanagement/marisk/marisk-in-der-fassung-vom-20-12-2005-598606 (Accessed: 2 Jan. 2024)
9. ISO/IEC 27000:2018 - Information security management systems. https://www.iso.org/standard/73906.html (Accessed: 8 Jan. 2024)
10. ISO/IEC 27001:2022 - Information security management systems. https://www.iso.org/standard/27001 (Accessed: 8 Jan. 2024)
11. ISO 31000:2018 - Risk Management. https://www.iso.org/standard/65694.html (Accessed: 8 Jan. 2024)
12. ISO 19600:2014 - Compliance management systems. https://www.iso.org/standard/62342.html (Accessed: 8 Jan. 2024)
13. ISO 37301:2021 - Compliance management systems. https://www.iso.org/standard/75080.html (Accessed: 8 Jan. 2024)
14. Basle Committee On Banking Supervision: Basel III: A global regulatory framework for more resilient banks and banking systems, Dec. 2010 and rev. June 2011
15. The General Data Protection Regulation (GDPR), the Data Protection Law Enforcement Directive and other rules concerning the protection of personal data. https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_en (Accessed: 8 Jan. 2024)
16. Corporate sustainability reporting. https://finance.ec.europa.eu/capital-markets-union-and-financial-markets/company-reporting-and-auditing/company-reporting/corporate-sustainability-reporting_en (Accessed: 8 Jan. 2024)
17. Global Reporting Initiative. globalreporting.org (Accessed: 8 Jan. 2024)
18. Lydenberg, S., Rogers, J., Wood, D.: Industry-Based Sustainability Reporting on Key Issues. Hauser Center of Harvard University (June 2010)

19. Sarbanes–Oxley Act of 2002 - Public Company Accounting Reform and Investor Protection Act. https://www.govinfo.gov/content/pkg/PLAW-107publ204/html/PLAW-107publ204.htm (Accessed: 6 Jan. 2024)
20. Karagiannis, D., Mylopoulos, J., Schwab, M.: Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act. RE 2007: 315–321. https://doi.org/10.1109/RE.2007.15
21. Fein, M.L.: Dodd-Frank Wall Street Reform and Consumer Protection Act. https://www.congress.gov/111/plaws/publ203/PLAW-111publ203.pdf (Accessed: 8 Jan. 2024)
22. The HIPAA Privacy Rule. hhs.gov (Accessed: 8 Jan. 2024)
23. Mahendra, I., Meyliana, M., Prabowo, H., Hidayanto, A.N.: Information Technology Challenges for Integrated Governance, Risk and Compliance (GRC). 2022 1st international conference on smart technology, applied informatics and engineering
24. Finma.ch: Rundschreiben 2023/1 Operationelle Risiken und Resilienz – Banken, 7.12.2022
25. Erlen, B., Isaak, A.J.: BWL-Kennzahlen: [Deutsch - Englisch], Jan. 2015
26. Fill, H., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. Enterp. Model. Inf. Syst. Archit. **8**(1) (2013)
27. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: representing knowledge about information systems. ACM Trans. Inf. Syst. **8**(4) (1990)
28. Karagiannis, D., Kühn, H.: Metamodeling Platforms. In EC-web 2002 Sep 2 (Vol. 2455, No. 2002, p. 182)
29. Karagiannis, D.: Agile modeling method engineering. Presented at the Proceedings of the 19th panhellenic conference on informatics (2015)
30. Vicente, P.C., Silva, M.M.D.: A Business Viewpoint for Integrated IT Governance, Risk and Compliance. 2011 IEEE World Congress on Services
31. Visic, N., Fill, H., Buchmann, R.A., Karagiannis, D.: A domain-specific language for modeling method definition: From requirements to grammar. IEEE 2015
32. Fill, H.-G., Gericke, A., Karagiannis, D., Winter, R.: Modellierung für Integrated Enterprise Balancing. Wirtschaftsinf. **49**(6), 419–429 (2007). https://doi.org/10.1007/s11576-007-0094-6
33. OBrien, J., Marakas, G.: Management Information Systems, 10th edn. McGraw/Hill (2007)
34. Strategy Layer: ArchiMate Specification. https://pubs.opengroup.org/architecture/archimate3-doc/ch-Strategy-Layer.html (Accessed: 8 Jan. 2024)
35. Nasif, Y.: How Composable Architecture Helps Leaders Master the Pace of Business Change. Gartner, Proceedings of IT Expo Barcelona, October 2023
36. BOC Group ADOGRC. boc-group.com/adogrc (Accessed: 2 Jan. 2024)
37. ONR 49000 Series - Risikomanagement für Organisationen und Systeme. https://www.austrian-standards.at/dokumente/produkte-loesungen/kostenlose-services/fachinformationen/06-risikomanagement/fachinformation06_risikomanagement.pdf (Accessed: 8 Jan. 2024)
38. Dimitris Karagiannis: A Business Process-Based Modelling Extension for Regulatory Compliance. Multikonferenz Wirtschaftsinformatik 2008. http://ibis.in.tum.de/mkwi08/17_IT-Risikomanagement_-_IT-Projekte_und_IT-Compliance/07_Karagiannis.pdf

Applying AMME for Constructing User-centric Services for Enterprise Architecture

Christoph Moser

Abstract This paper addresses the challenge of democratizing enterprise architecture by improving the accessibility of established modeling methods such as ArchiMate to a wider audience. Considering that the complexity of such methods can lead to misinterpretation and resistance from users who lack expert knowledge in modeling, the paper argues for a solution based on Agile Modelling Method Engineering and Situational Method Engineering. The main goal is to empower users to actively participate in the design process and to encourage collaboration.

Keywords Agile modeling method engineering · AMME · Situational method engineering · SME · Enterprise architecture · Roadmapping · ArchiMate

1 Introduction

An enterprise is a product of human creation. It is organized and given vitality by individuals with a common goal. Establishing enterprises entails numerous design decisions made by subject matter experts who compose various core elements of an enterprise such as enterprise goals, business processes and underlying technologies. To this end numerous modeling methods have become established in the field of enterprise modeling. Prominent examples are ArchiMate [1], Zachman [2], UML [3], capability-oriented approaches [4] and of course the wide range of process management methods such as BPMN [5]. These methods differ both in their degree of abstraction and in their perspectives on the organization.

Within the European information systems research community, a specialized community in method engineering (ME) has emerged, dedicating its focus to method development [6]. More than two decades ago, this ME community recog-

C. Moser (✉)
BOC Group, Vienna, Austria
e-mail: christoph.moser@boc-group.com

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_9

nized the necessity for adaptable methods, understanding that a "one-size-fits-all" approach is unsuitable (see e.g. [6–11]).

In this paper we focus on methods in context of Enterprise Architecture (EA). One prominent method is TOGAF's Architecture Development Method [12]. It offers a comprehensive guide for implementing EA initiatives, but lacks detailed guidance for the creation of specific deliverables such as capability maps, architecture assessments, technology radars and capability-based roadmaps. While ArchiMate [1], a widely used modeling language for EA, offers recommendations for viewpoints (i.e. modelling patterns) tailored to specific EA tasks, it does not provide instructions on creating and using views based on these patterns.

Users of such methods and frameworks oftentimes are forced to grasp the entire method and comprehend all its concepts, to make use of them for their individual needs. Alternatively, a subject matter expert may need to instruct them on the relevant parts for effective utilization. This prerequisite has a negative impact and may discourage users from using the methods. In addition, the use of these methods by different teams or team members can lead to inhomogeneous results and model inconsistencies. This is true both across organizations and, even worse, within a project. This paper provides a solution proposal to these problems.

The remainder of this paper is structured as follows. Section 2 delves into an in-depth exploration of the problem statement, presents background information on foundations and related work, and introduces a recurring example that is referred to throughout the paper. In Sect. 3, the creation of User-centric Services for EA is detailed, following an established procedure model for Agile Modeling Method Engineering. The concluding Sect. 4 evaluates the effectiveness of the UCS concept by means of an empirical experiment and presents the results in a conclusive SWOT analysis.

2 Problem Statement, Background and Continuous Example

2.1 Problem Statement

One challenge in enterprise architecture is the inherent limitation of "one-size-fits-all" methods, which prove ineffective in guiding method users to produce the required deliverables. These methods pose a two-fold challenge: (1) They force non-EA experts to invest time in understanding the entire method, often overwhelming them, when in fact they only need parts of the method. This learning curve may **discourage active participation**. (2) The modeling freedom offered by these methods means that different teams or even team members generate results with different structures and levels of detail. This lack of standardized output makes EA analysis across different teams difficult and **prevents enterprise-wide insights into the EA**.

To address these issues we propose the concept of User-centric Services (UCS). UCS can be understood as situative methods based on the well-known EA standard ArchiMate. Examples of such User-centric Services are:

- Capability Assessment: A service that evaluates the strategic importance and maturity of business capabilities.
- Capability-based Roadmapping: A service that helps define, prioritize and plan strategic-level requirements and investments.
- Application Investment Planning: A service that assesses software applications based on the TIME model [12].
- Architecture Threat Modeling: A service that identifies threats at the business, application, and technology layers of an EA.
- Technology Trend Scouting: A service that identifies and communicates emerging technology trends in the form of a technology radar.

The concept of UCS is based on the foundational characteristics of Agile Method Engineering as defined in [13]:

- ADAPTABILITY: ability to change existing concepts;
- EXTENSIBILITY: ability to add new concepts;
- INTEGRABILITY: ability to integrate concepts;
- OPERABILITY: ability to offer functionality for model operations (e.g., automated model visualization, validation checks, simulation);
- USABILITY: the ability to guarantee ease of use and model comprehensibility.

2.2 Foundations and Related Work

Method components. To define the UCS method, we employ a recognized framework [14] which structures modeling methods into the following three components:

- **Modeling Language:** The modeling language establishes the language alphabet, including notation, grammar, and machine-readable semantics. In the field of EA, there are various modeling languages, with the Zachman Framework [2] being the first schema for organizing architectural artifacts. One of the most widespread and comprehensive frameworks is ArchiMate [1], known for its globally recognized and well-specified modeling language.
- **Mechanisms & Algorithms:** In essence, mechanisms and algorithms involve the collection, analysis, and visualization of architectural data, described by the modeling language. Examples include decision support mechanisms based on simulation or AI-based techniques, as well as the automated generation and filtering of diagrams (i.e. views).
- **Modeling Procedure:** The modeling procedure defines the steps that modelers must take to achieve a modeling goal. Often, as in ArchiMate, the modeling procedure is not explicitly defined.

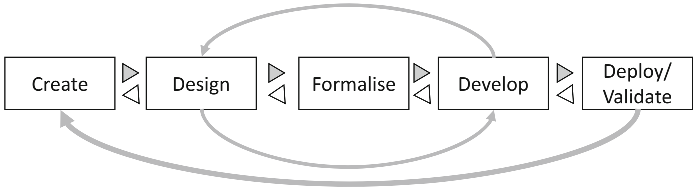


Fig. 1 AMME conceptualization lifecycle [13]

Agile modeling method engineering. Agile modeling method engineering (AMME) [13] is based on a metamodeling approach designed to identify, design and implement modeling requirements. It guides the design and adaptation of modeling methods through a 5-step approach (Fig. 1), encompassing considerations from the design phase to the methods actual deployment and validation. We utilize AMME for defining the UCS.

Situational Method Engineering. Situational Method Engineering is an adaptive approach in information systems and software engineering, tailoring methods to specific organizational contexts. The construction of situational methods involves the initial selection of method components, referred to as method fragments or chunks, that can be customized and combined to create tailored methods for specific projects or contexts. The methodbase is a centralized repository that stores and provides access to reusable method fragments. Thus, the methodbase promotes reusability and efficiency.

The origin of these fragments is not a crucial factor; they can be derived from existing methods, adhere to standardized metamodels [15] (in our case ArchiMate) or build in an ad-hoc manner in case no suitable fragments were available in the methodbase.

ArchiMate. ArchiMate [1] is one of the mentioned "one-size-fits-all methods". It is a widely used modeling language for EA, providing a standardized and well-specified framework for representing and visualizing the structure and behavior of an organization's architecture in the form of the typical node/edge diagrams. It encompasses more than 60 object types (i.e. elements) such as *business processes*, *application components*, *system software*, and *business actor* to accurately depict the structure and dynamics of an organization. Its 11 relationship types cover a wide range of connections. The list of relationship types includes structural relationships like *composition* and *aggregation*, dependency relationships that capture the dependencies between elements (e.g. *access*), and dynamic relationships that represent the behavior and interaction between elements (e.g. *trigger*) [1].

The combination possibilities of its numerous element types and relationships introduces complexity, necessitating the method engineers to find a delicate equilibrium between offering adequate detail and to communicate it to the method users. To make the method easier, ArchiMate recommends the use of *viewpoints*, i.e. subsets of ArchiMate elements and relationships that allow users to model specific aspects of an enterprise architecture, addressing stakeholder concerns and expressing them in diagrams. These *viewpoints* serve as patterns for the method users. On the one hand, they enable the uniform modeling of relevant parts of the organization. On the other hand, they allow the method users to focus on a smaller set of *concepts*. See for example [16] who offers a comprehensive set of such patterns in his ArchiMate Cookbook.

2.3 Continuous Example: UCS Capability-Based Roadmaps

Capability-based Planning (CBP) approaches provide a solution for aligning an organization's capabilities with its strategic objectives [17]. It is a well-known tool from the field of EA. Enterprise Architects can utilize CBP to model the current and desired abilities of an organization at a higher level of abstraction, taking into account the organization's strategy and its environment. In essence, CBP provides two main deliverables:

- a catalogue of assessed capabilities usually depicted in the form of a capability map and
- a corresponding roadmap that outlines prioritized requirements assigned to specific implementation periods (e.g. quarters of the year).

Moving forward, we will focus on the second deliverable, namely on Capability-based Roadmaps. We employ this use case to introduce our concept of UCS and explain, based on this example, how to proceed in general when building UCS.

3 Defining User-Centric Services for EA

In this chapter, we describe the procedure for creating User-centric Services based on AMME's procedure model and principles [13]. The chapter is structured into the phases of AMME.

3.1 Create

In the initial phase of method engineering, stakeholder identification and analysis are crucial, emphasizing the discovery of methods that cover specific user requirements and contribute to achieving the anticipated business outcomes. Gathering user requirements involves engaging potential users in interviews or workshops to identify methods that solve the current business problem and to uncover limitations of their use based on contingency factors such as the skill set of the participants and the available time.

At the end of this phase, the objectives for which the method is applied, the requirements of the method users, suggestions for the use of one or more existing methods, and proposals for the adaptation of such methods are defined.

Considering our continuous example, stakeholder roles include enterprise architects and business experts, the latter lacking proficiency in ArchiMate. Despite ArchiMate meeting all requirements for capability-based roadmapping, its complexity may be overwhelming. Thus one major requirement is to optimize the method for non-modeling-experienced business experts who are uncomfortable with the common node/edge diagrams used in ArchiMate when it comes to the definition of roadmaps.

3.2 Design

The design phase involves designing and specifying the building blocks of the modeling method, including language, mechanisms, and procedures. This is done by "reengineering" the anticipated deliverable, identifying the necessary steps, data requirements, and supporting mechanisms and algorithms required to produce the deliverable.

To put it differently, each step in the modeling procedure of a method can be seen as a combination of method fragments in SME [18]. Each step consists of method fragments of type "product" and "process":

- *Product fragments* include the necessary metamodel and the resulting output, typically in the form of a diagram or editor.
- *Process fragments* encompass the steps that provide guidance, along with the necessary mechanisms and algorithms that partially or fully automate each step.

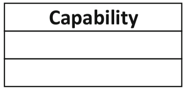
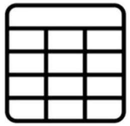
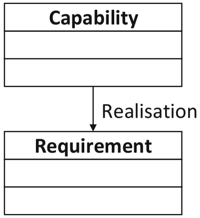
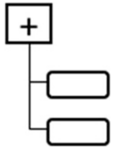
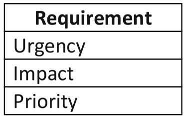
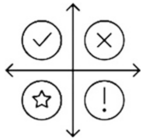
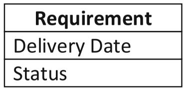
Table 1 describes the fragments required for our continuous example.

The AMME principle of **EXTENSIBILITY**, the ability to add new concepts, is fulfilled by the situational nature of the UCS. Continuous development of new method fragments is supported by adding process and product fragments to the methodbase.

3.3 Formalize

Formalization is required to build-up the methodbase and to make it accessible. The process and product fragments must be described in such a way that they can be found and combined. This means that method engineers specify the method fragments precisely to make possible identification of compatible product fragments and process fragments and thus to ensure "***OPERABILITY****: the ability to provide functionality for operating on models*" [13]. Another important AMME principle is **INTEGRATABILITY** to ensure that two consecutive steps of the UCS fit together.

Table 1 Method fragments for UCS capability-based roadmap

Process Fragments		**Product** Fragments	
UCS Step (part of Modeling Procedure)	Mechanism & Algorithm	Metamodel (snippet from ArchiMate)	Editor & Diagram
Select scope: Filter/select or create capabilities that are in scope.	Filter the model/repository for objects based on type and attribute.		Filterable table 
Create requirements: Assign requirements to capabilities.	Create objects and relations.		Collapsible tree 
Prioritize requirements: Use a prioritization method to prioritize the requirements based on input attributes "Urgency" and "Impact".	Set attribute values and calculate output attribute values.		Prioritization matrix 
Plan and track requirements: Assign the requirements to the time units (e.g. quarters) and track their degree of completion (i.e. status).	Set attribute value.		Kanban board with timeline 

It is defined at language level by defining input-output relationships between two steps.

The "Mechanisms & Algorithms" and their interplay with the "Editors & Diagrams" can be declared in the form of mathematical expressions. In terms of style, our mathematical expressions are inspired by the FDMM formalism [19] which is mentioned as an example in AMME [13]. Below, we demonstrate how the function declarations for the process fragments of type "Mechanism & Algorithm" are defined for our continuous example. For the required description of the product fragments "Metamodel", we refer to the specifications in FDMM [19] due to space constraints.

Take the **first step** from Table 1. During this step, the user of the method will chose relevant objects of type "Capability" from the repository. This step necessitates the use of a filter function to selectively extract the relevant objects from the repository/model.

The required filter function is declared as follows:

$$\text{ObjectSelection} : O \rightarrow \{o \in O \mid o.\text{ObjectType} = \text{objectType} \wedge o.\text{Attribute1} \\ = \text{value1} \wedge o.\text{Attribute2} = \text{value2} \wedge \ldots\}$$

where O is the set of objects residing in the repository/model.

In our continuous example we select objects of type "Capability" e.g. with attribute "level" = 1.

$$\text{ObjectSelection} : O \rightarrow \left\{o \in O \mid o.\text{objectType} = \text{``Capability''} \wedge o.\text{Level} = \text{``1''}\right\}$$

In the **second step** of our continuous example, we utilize a function for creating and connecting objects of type "Requirement" with the previously selected "Capabilities". This assignment is made via relationships of type "Realisation". The function is declared as follows:

$$\text{AssignSubordinatedObject} : O \rightarrow \left(R^T \times \left(O^T \times O^{NEW}\right)\right)$$

where

O is the set of objects from which assignments are made
R^T is the type of relationship that is used
O^T is the type of the objects that are created and connected and
O^{NEW} represents the set of new objects.

In our continuous example O is the set of selected capabilities (step 1), O^{NEW} is the set of new requirements, O^T is the ArchiMate object type "Requirement" and R^T is the relationship type that connects capabilities with requirements.

In the **third step**, the UCS leverages a 2×2 prioritization matrix of type "Eisenhower". Eisenhower Matrix is a prioritization framework that categorizes requirements based on their urgency and importance to aid prioritization of workloads [20]. For this step, a graphical editor that displays four distinct fields on a canvas is required. Requirements will be strategically placed on the canvas by the method users and consequently assigned to one of the matrix fields: “do later,” “do first,” “eliminate,” or “reconsider.” The underlying function enabling this matrix editor is declared as follows:

$$\text{EisenhowerMatrix} : O \rightarrow \text{Priority}$$

where O is a set of objects with an associated attribute “Priority” and Priority $\rightarrow$ (Impact$\times$Urgency) and Priority $=$ {do later, do first, eliminate, reconsider}

As a reminder: In the metamodel part, there must be object types that satisfy these requirements. In the case of our continuous example, the object type “Requirement” requires the attributes “Impact”, “Urgency”, and “Priority”.

In the **fourth step**, our UCS utilizes a Kanban Board editor to strategically allocate the requirements to designated quarters for planned delivery. Additionally, a status value is assigned to each requirement. The underlying function, that represents the features of the Kanban Board, is declared as follows:

$$\text{KanbanBoard} : O \rightarrow (\text{PlannedDeliveryDate} \times \text{Status})$$

where O is a set of objects with attributes “PlannedDeliveryDate” and “Status” and PlannedDeliveryDate is a date and Status $=$ {completed, caution, on-track, off-track, not started, out of scope}

3.4 Develop

For the creation of the functional UCS we use the metamodeling platform ADOxx. Metamodeling platforms provide pre-existing generic functionality, such as model storage, filter/search capabilities, and configuration options for the metamodel. By utilizing these built-in features, method engineers can streamline the process and concentrate on constructing the core components of the method. We utilize ADOxx (1) to populate and manage the UCS methodbase, as well as (2) to implement UCS templates like *Capability-based Roadmap* from our continuous example.

Here, we benefit from another advantage: ADOIT, a widely used EA suite [21], also based on ADOxx, already comes with numerous features and, in particular, with the ArchiMate metamodel. Thus, our methodbase cannot only be built with the features of ADOxx but can also leverage features from ADOIT. To make the implementation of UCS templates possible, ADOxx has been expanded with the UCS framework that enables the configuration of UCS templates (so-called *workspaces*) by combining method fragments from the methodbase.

In Fig. 2, we can see the UCS of our continuous example “Capability-based Roadmapping.” The top area of the screenshot displays the steps that guide the

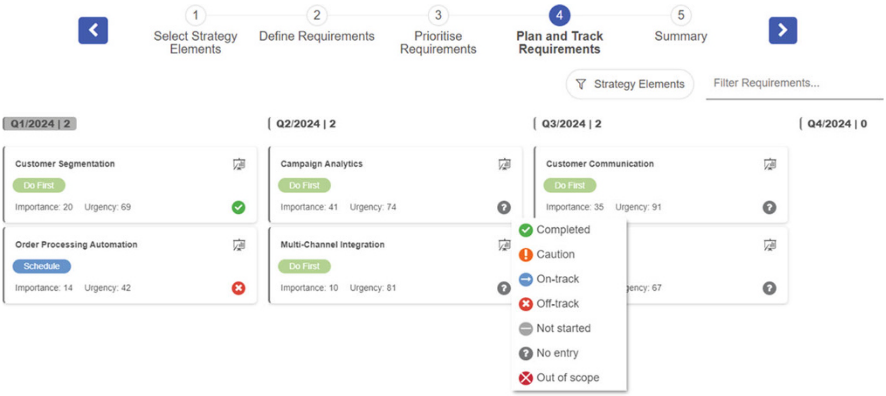


Fig. 2 UCS Capability-based Roadmap, Step 4 "Plan and Track Requirements"

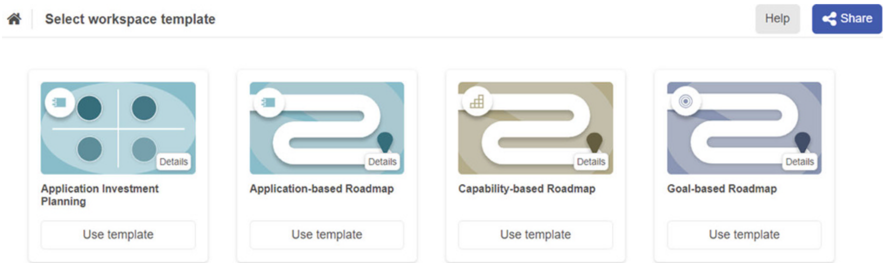


Fig. 3 Workspace template selection—creation of a workspace based on a UCS template

method user to the expected outcome. Notably, step 4 is activated, revealing the kanban board which facilitates the editing and visualization of the roadmap.

3.5 Deploy/Validate

In the deploy/validate phase, the UCS template is assessed in a real-world setting to determine its effectiveness and suitability. For this purpose, ADOIT (based on ADOxx) is used as a runtime. Starting from the UCS template, a UCS instance, a so-called *workspace*, is instantiated. The workspace includes all the described steps and the associated editors. Users can directly conduct the modeling within it. Figure 3 illustrates a selection dialog of various workspace templates including the UCS for Capability-based Roadmaps. An example of one of its workspace steps was already shown in Fig. 2.

The evaluation phase emphasizes the importance of "*USABILITY, the ability to deliver satisfactory user interaction and model comprehensibility*" [13]. The feedback obtained in this phase may lead to setbacks in previous phases.

4 Evaluation and Conclusive SWOT Analysis

The effectiveness of the UCS approach was examined through an empirical experiment. In this section, we describe the setup of the experiment and its results.

4.1 Experiment Setting

Objective. The experiment aimed to assess and compare the efficiency and user experience of two software solutions for capability-based roadmapping: traditional ArchiMate modeling (using node-edge diagrams and graphical modeling) and the UCS Capability-based Roadmap. The results are meant to infer the performance comparison between UCS and traditional ArchiMate modeling in general. In the experiment, both approaches were examined based on a modeling exercise.

Participants. The participants, university students in the 3th semester with a general understanding of enterprise modeling and basic experience in BPMN, EPC, Entity-Relationship diagrams, and PESTLEweb, received a one-hour introduction to ArchiMate before the experiment. 20 participants took part in the experiment.

Experimental Design. The experiment consisted of two parts. In the initial part, participants were divided into two groups, each assigned to one of the software solutions (traditional modeling or UCS Capability-based Roadmap). Both groups received the same assignment. Two weeks later, the groups switched tools, so that in the end all participants evaluated both approaches. Results from (two) students who only participated in one part of the experiment were not evaluated.

Tasks. Participants were assigned the task of modeling a capability-based roadmap. For details on the specific task, please refer to the appendix.

Measures. The experiment focused on three key metrics:

- Completion time: The duration participants took to complete the task.
- Quality of the models: Accuracy and completeness of the created roadmaps.
- Mental effort: The mental load experienced by the participants while executing the task.

4.2 Results and Implications

Given the limited participant pool, the analyses exclusively rely on basic statistical measures, including average and median. The ensuing sections present and discuss the outcomes:

Completion time. The differences in the processing time of the task appear significant. Figure 4 displays the average processing time and the median processing time for both approaches. The processing time to complete the exercise in UCS is, on average, ~28% shorter.

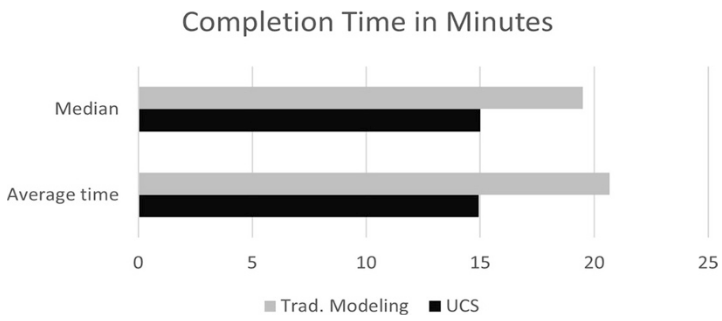


Fig. 4 Completion time is lower in UCS

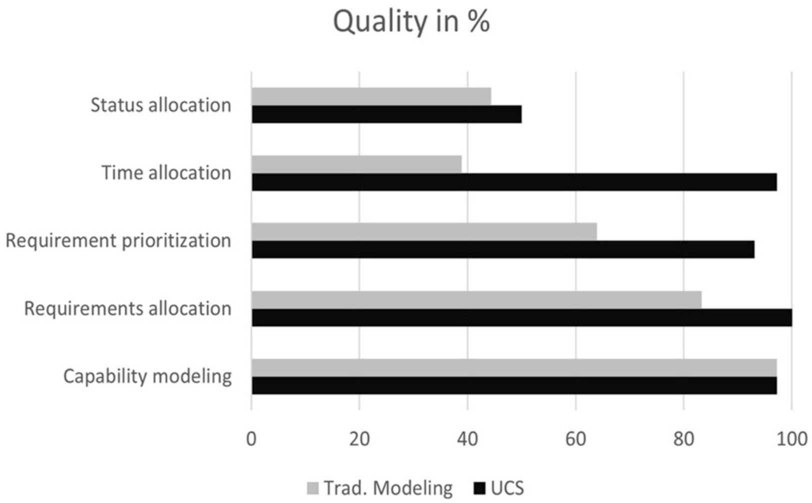


Fig. 5 Quality of models is higher in UCS

Quality of the models. The accuracy of the models is evaluated for the subtasks (1) Capability modeling, (2) Requirement allocation, (3) Requirement prioritization, (4a) Requirement scheduling, and (4b) Status allocation (see Fig. 5).

Each subtask is assessed on a scale of up to 2 points, with the scoring criteria as follows: task not or badly executed (0 points), minimal errors but correct solution path identified (1 point), and error-free (2 points). Each task was assessed individually. Potential subsequent errors were not specifically addressed.

In each step, UCS consistently delivers more accurate results. The only exception is "Capability Modeling" where both approaches achieve almost 100% accuracy. The superior performance of UCS in "Requirements Allocation" is attributed to the fact that in traditional modeling, wrong relationships were at times used due to the degree of freedom offered by the modeling editor. In UCS, the defined relationship type was automatically set, eliminating the risk of incorrect choices.

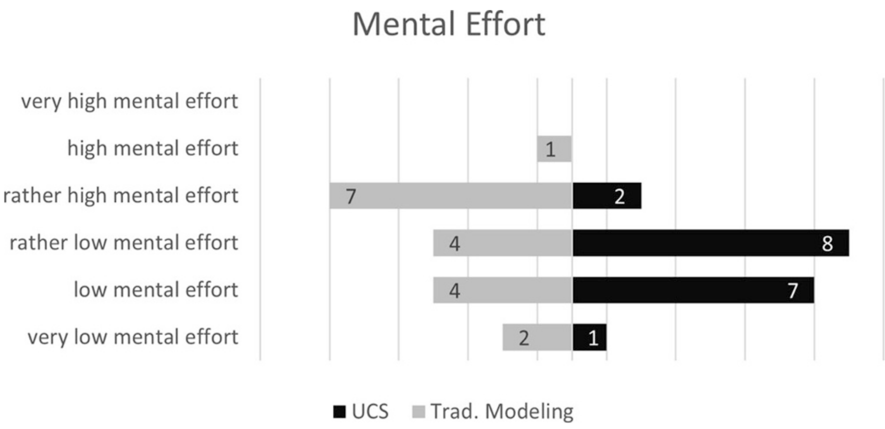


Fig. 6 Mental effort is lower in UCS

In “Requirements Prioritization” and “Time Allocation” the UCS’s task-specific editors excel over the conventional property boxes used for data input in traditional modeling. The visual presentation of results in UCS mitigates the risk of missing to enter appropriate values and makes visual verification simpler. Notably, “Status Allocation” performed significantly worse in UCS compared to the other UCS tasks, with participants overlooking the task due to its somewhat hidden placement in the user interface. This oversight can be attributed to participants relying on insufficient UCS guidance. It highlights a User Experience (UX) flaw that could have been avoided through a dedicated UCS step, for instance.

Mental effort. The mental effort resulting from the execution of the task was rated by the participants on a scale of 6 values from “very high mental effort” to “very low mental effort”. The question was adopted from a similarly designed experiment on the topic of “multimedia learning” [22]. The results demonstrate a superior performance of the UCS approach (Fig. 6).

In summary, it can be concluded that the User-centric Service approach outperforms the traditional modeling approach across all evaluated measures: completion time, quality of models, and mental effort. However, to statistically bolster this finding more experiments need to be conducted, as discussed in the “threats” section in the following section.

4.3 Conclusive SWOT Analysis

In the conclusive SWOT analysis, the evaluation of the UCS method is summarized and some ideas for future research are shared.

- **Strengths:** By applying the AMME method, User-centric Services can be systematically developed and refined. The conducted experiment proves that the

UCS approach outperforms the traditional node/edge-based ArchiMate modeling.

- **Weaknesses:** While there are considerations for additional UCS and how concrete method fragments can be reused, these UCS and the fragments need to be specified/implemented. This is particularly important to better estimate the level and extent of reuse of the method fragments and will be subject to future research.
- **Opportunities:** Considering that, in theory, fragments can exist at any granularity [11], one could argue that combinations of UCS can be conceptualized as higher-level services. For instance, combining a UCS Capability Assessment and a UCS Capability-based Roadmap could form an overarching solution, referred to as "Capability-based Planning". Consequently, UCS could themselves be described within a methodbase and combined according to a set of rules.
- **Threats:** The experiment was conducted based on a prototypical UCS. The hypothesis that UCS outperforms the traditional modeling approach has been proven. However, further experiments with additional UCS on topics such as Application Investment Planning, Capability Analysis, Business Continuity Analysis, etc., need to be conducted to confirm the hypothesis also for other UCS. This is the task of future research.

Appendix

Task given to the participants of the experiment:

- Please read the entire document before you start modeling!
- Record the start and end times. The start time should coincide with the moment you begin reading.
- Fill-in the section "Feedback questions – AFTER CONDUCTING THE EXPERIMENT"

Your Task

You are part of the management team at a toy manufacturing company and aim to develop a roadmap for the enhancement of the "Marketing" and "Order Management" capabilities by implementing the following improvements:

Marketing

- Customer Segmentation: Define and implement a system for categorizing customers based on demographics, behaviors, and preferences to tailor marketing strategies effectively.
- Campaign Analytics: Implement analytics tools to measure the performance of marketing campaigns, track customer engagement, and assess the return on investment.
- Multi-Channel Integration: Ensure seamless integration across various marketing channels, such as social media, email, and traditional advertising, to provide a consistent brand experience.

Order Management

- Inventory Visibility: Establish a system that provides real-time visibility into inventory levels, ensuring accurate tracking of product availability and preventing stockouts.
- Order Processing Automation: Implement automation in order processing to streamline workflows, reduce errors, and enhance the efficiency of fulfilling customer orders.
- Customer Communication: Develop a communication system that keeps customers informed about order status, shipping details, and any potential delays, fostering transparency and customer satisfaction.

Create a roadmap to enhance the 'Marketing' capability, taking into account the above requirements of customer segmentation, campaign analytics, and multi-channel integration. Also consider 'Order Management' and the above listed challenges in your roadmap.

All of the items in your roadmap have an urgency score higher than 50 and an importance score higher than 50.

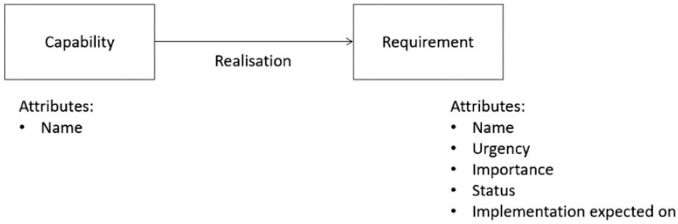
Assign a deliverable date to the planned changes.

- "Customer Segmentation" and "Order Processing Automation" need to be implemented by end of Q1/2024.
- "Campaign Analytics" and "Multi-Channel Integration" need to be implemented by end of Q2/2024.
- All other changes/requirements need to be implemented at latest by end of Q3/2024.

Now imagine that Q1/2024 passed by. The tasks "Customer Segmentation" was completed successfully. The task "Order Processing Automation" has not been completed and has status "Off-track".

Additional Input

Use ArchiMate as a modeling language. Required viewpoint (metamodel):



Feedback Questions: AFTER CONDUCTING THE EXPERIMENT

Start time of experiment:————
End time of experiment: ————
In solving the tasks I invested

[] very low mental effort
[] low mental effort
[] rather low mental effort
[] rather high mental effort
[] high mental effort
[] very high mental effort

References

1. The Open Group: ArchiMate® 3.2 Specification. https://pubs.opengroup.org/architecture/archimate3-doc/
2. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. **26**, 276–292 (1987)
3. OMG: OMG® Unified Modeling Language® (OMG UML®). https://www.omg.org/spec/UML/ (2017)
4. Zdravkovic, J., Stirna, J., Kuhr, J.-C., Koç, H.: Requirements engineering for capability driven development. Presented at The Practice of Enterprise Modeling: 7th IFIP WG 8.1 Working Conference, PoEM 2014, Manchester, November 12–13, 2014. Proceedings 7 (2014)
5. Object Management Group (OMG): Business Process Model and Notation (BPMN). https://www.omg.org/spec/BPMN/2.0/PDF (2011)

6. Gericke, A., Fill, H.-G., Karagiannis, D., Winter, R.: Situational method engineering for governance, risk and compliance information systems. Presented at the Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (2009)
7. Brooks, F.P., Bullet, N.S.: Essence and accidents of software engineering. IEEE Comput. **20**, 10–19 (1987)
8. Avison, D.E., Wood-Harper, A.T.: Information systems development research: an exploration of ideas in practice. Comput. J. **34**, 98–112 (1991)
9. Ter Hofstede, A.H., Verhoef, T.: On the feasibility of situational method engineering. Inf. Syst. **22**, 401–422 (1997)
10. Glass, R.L.: Matching methodology to problem domain. Commun. ACM. **47**, 19–21 (2004)
11. Ralyté, J.: Towards situational methods for information systems development: engineering reusable method chunks. Presented at the Proceedings of the 13th International Conference on Information Systems Development. Advances in Theory, Practice and Education (2004)
12. Gartner: Use TIME to Engage the Business for Application and Product Portfolio Triage (2022)
13. Karagiannis, D.: Agile modeling method engineering. Presented at the Proceedings of the 19th panhellenic conference on informatics (2015)
14. Karagiannis, D., Kühn, H.: Metamodelling platformss. Presented at the EC-web (2002)
15. Henderson-Sellers, B., Gonzalez-Perez, C., Ralyté, J.: Comparison of method chunks and method fragments for situational method engineering. Presented at the 19th Australian Conference on Software Engineering (aswec 2008) (2008)
16. Hhosiaisluoma, E.: ArchiMate Cookbook 1.0. https://www.hosiaisluoma.fi/ArchiMate-Cookbook.pdf (2019)
17. Papazoglou, A.: Capability-based planning with TOGAF® and ArchiMate® (2014)
18. Mirbel, I., Ralyté, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. Requir. Eng. **11**, 58–78 (2006)
19. Fill, H.-G., Redmond, T., Karagiannis, D.: FDMM: a Formalism for Describing ADOxx Meta Models and Models (2012)
20. Čiarnienė, R., Vienažindienė, M.: The conceptual model of time management. Presented at the Book of Proceedings (2014)
21. Gartner: Critical Capabilities for Enterprise Architecture Tools. https://www.gartner.com/document/4953731?ref=solrAll&refval=386735708& (2023)
22. Paas, F., Ayres, P., Pachman, M.: Assessment of cognitive load in multimedia learning. Recent Innov. Educ. Technol. Facil. Stud. Learn. Inf. Age Publ. Inc Charlotte NC. 11–35 (2008)

Applying the Powerful Concept of Meta-Models

Martin Nemetz

Abstract During my university studies I got acquainted with the concept of meta-modelling. While at first, it appears to be complex, meta-models and their application turn out to be powerful for structuring and solving problems. In this chapter, I aim to demonstrate to the interested reader that meta-modelling is not complex at all and that it can be applied rather intuitively. With the example of traffic signs, I will try to demonstrate the power of meta-models. Later, I will provide a scientific example of how to use meta-models for making Intellectual Capital Reports comparable. Finally, I will present two use cases from industry – one showing (once more) the power of meta-models, the other one describing limitations in their application that may require further research.

Keywords Metamodeling · Intellectual capital · Reporting · Artificial intelligence · Software industry

1 Introduction and Reflection of the Two-Decade Long Work with Prof. Karagiannis

It was in October 2002 that I have met Prof. Karagiannis for the first time. Actually, it was not directly meeting him, but rather *experiencing* him. He gave the introductory lecture on *"Business Process Management"* [1] during my course of studies of *International Business Administration*. I remember I was impressed. He demonstrated such a different style of lecturing compared to all other professors and lecturers I have met before at the University of Vienna. Prof. Karagiannis spoke without showing slides (or what some professors still used back then, overhead transparencies). Rather, he sketched on the whiteboard and has explained to us the importance of Information Technology (IT) for companies and their processes. He introduced this topic to us as students in a practical way. No formulae, no

M. Nemetz (✉)
Buchs, Switzerland

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_10

theories, no hypotheses. It has been a 90-min deep-dive into how IT has become the *blood circulation* of modern enterprises. However, Prof. Karagiannis did not paint a rosy picture. He told us that it requires studying and hard work for getting to experience all this in a job. This was the moment that I got clarity for myself of what I want to do professionally: I was eager to work in the intersection of IT and Business Administration. In University terms, this relates to a subject named *Business Informatics*.

Consequently, I have registered for all respective lectures. And in February 2004, I was going for my last exam before graduation covering the full scope of what we have learnt about Business Informatics (coincidentally, the assistant assessor next to Prof. Karagiannis, was Hans-Georg Fill, one of the co-editors of this *liber amicorum*, meanwhile professor at Université de Fribourg and a friend). While studying the subject of Business Informatics, I have worked in parallel at Accenture and I got to a glimpse of the practical experience of what it means to work in IT projects for companies. I have enjoyed this time a lot, but I wanted to get a more theoretical foundation. Hence, I have asked Prof. Karagiannis right after having passed the final exam whether he would accept me as a PhD candidate. I consider it my luck that he took the risk to accept me in his team, as we have not worked together before. Like what I have experienced in my first lecture with him, I got to experience during my PhD studies: study hard, deep-dive into literature and scientific methods, but also apply the findings in practice and find evidence whether those make a difference in reality. In other words: perform *Design Science* [2]. For me personally, this felt and continuously feels to be a source of interesting research work.

After having completed my PhD studies in 2008, I have started to work for an international company abroad. However, Prof. Karagiannis and I have stayed in touch, and we have worked together on various occasions. And I am glad and grateful that I can call Prof. Karagiannis to be my mentor for almost 20 years now. More than once he has provided me with his view when it came to difficult personal or professional decisions.

The remainder of this book chapter is organized as follows: Sect. 2 provides an introduction into meta-modelling and ways how to explain it. Section 3 offers a short overview of how the concept of meta-models can be applied in the domain of *Intellectual Capital Management and Reporting* [3], before in Sect. 4 one example of and one difficulty in use case implementations of meta-modelling concepts are being presented. Section 5 provides an outlook and raises the research question how artificial intelligence could support meta-modelling concepts for documenting complex IT and software solutions in companies. Finally, this book chapter closes with acknowledgements.

2 Meta-Modelling and How to Explain It

Prof. Karagiannis heads the Department of Knowledge Engineering (DKE) at the University of Vienna. One of the foundations of his research group is the concept of meta-modelling [4, 5] and its application in various domains [6].

A Meta-model is defined as the model of a modelling language [4] and a modelling language in turn *"contains the elements with which a model can be described (. . .) by its syntax, semantics and notation*" [4]. A second element next to the modelling language is the modelling procedure describing application steps of the former to achieve results. Both the language and procedure can be summarized as modelling technique [4]. Finally, mechanisms and algorithms work on the (meta-)models described by the modelling language. These three elements (modelling language, modelling procedure, and mechanisms and algorithms) form the concept of a (meta-)modelling method [4]. Figure 1 depicts these components accordingly.

Based on practical experience, meta-models and their application require explanation for ensuring the audience understands the concept and eventually can apply it. In the following paragraphs, a use case shall help to introduce the concept accordingly. Figure 2 supports this explanation.

Meta-models and models stand in a hierarchical relation to each other [7]. In fact, the modelling language for creating a meta-model is called meta-modelling language. The language for creating a meta-meta-model (or also meta^{2}model) is called a meta-meta-modelling language (or meta^{2}modelling language) and so forth [4, 7]. Theoretically, one could add an infinite number of layers, however in practical terms it has proven that the meta^{2}model layer is typically covering the necessary scope and further levels are hardly applied (e.g. [8] or [9]). This leads to a four-layered language system for meta-modelling containing the layers (from bottom to top): instance, model, meta-model, and meta^{2}model [7]. When explaining this four-layered language system for meta-modelling for the first time to students or colleagues in companies, it oftentimes turns out to be difficult (a) to understand it and (b) being enabled to apply it. As mentioned above, an introduction of a use case may be helpful for achieving (a) and (b). The use case at hand looks at traffic signs as they are being used in Central Europe. On the lowest layer, the instance level, one starts with real traffic signs, i.e. a traffic sign being placed on the roadside, a traffic sign that one can see and touch. An example for such a traffic sign is the one indicating a *pedestrian crossing* at Feldkircher Strasse 81 in Schaan, Liechtenstein (in proximity to my workplace). Another instance of a traffic sign is the one indicating *no entry* placed in between the train station and bus terminal in Buchs SG, Switzerland (close to where I live).

The second layer depicts the respective models of these two traffic signs. The first one (pedestrian crossing) has got a rectangular shape and is mostly blue colored, the second one (no entry) is shaped as a circle and has a red line at its outer boarder. There is a third model next to the first two (rectangular and circular), which has got a triangular shape and has got again a red line at its outer boarder. These three models of traffic signs are called *Gebotszeichen* (rectangular, blue), *Verbotszeichen*

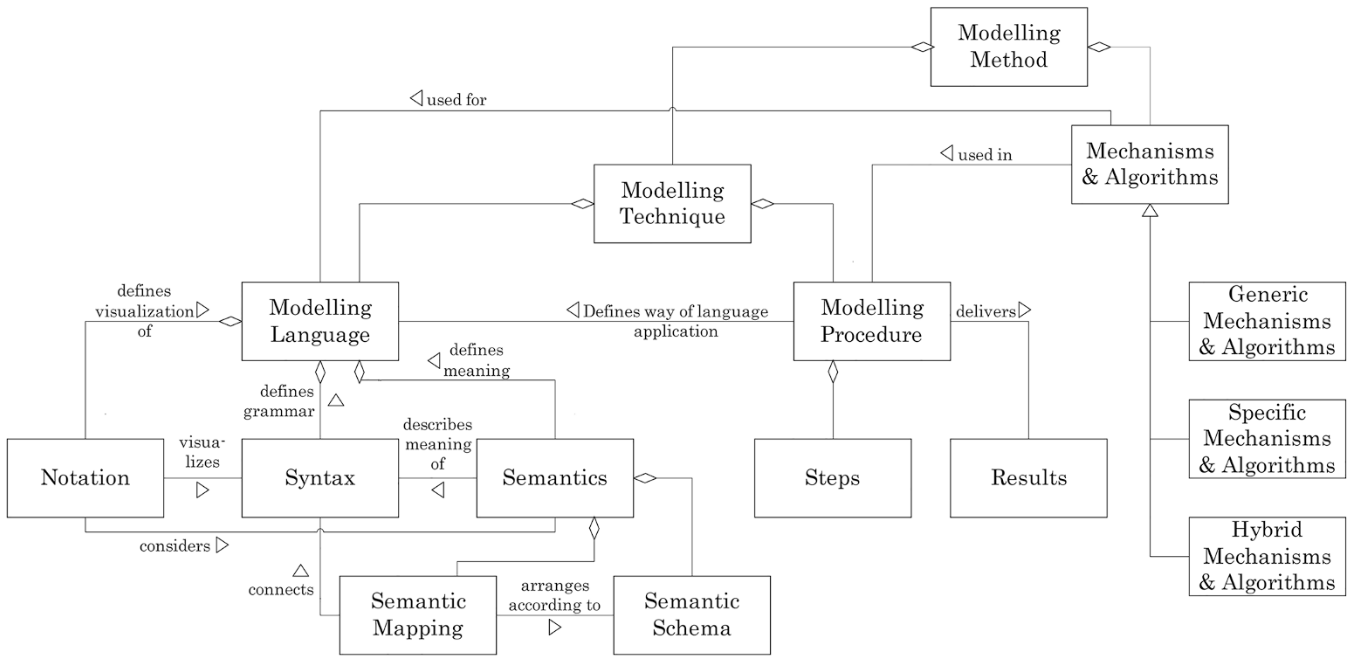


Fig. 1 Components of a (meta-)modelling method [4]

Fig. 2 Using traffic signs to explain the concept of meta-models

(circular, red line on the outer boarder), and *Gefahrenzeichen*[1] (triangular, red line on the outer boarder). These three models can each be instantiated according to actual needs as e.g. the *Gebotszeichen* can be placed next to a pedestrian crossing, then it would depict a man crossing a street, but it could also be instantiated for indicating a path walk, then it would be still rectangular and blue-colored but depict a walking man and child. Similarly, this can and is applied throughout Central Europe for the *Verbotszeichen* and the *Gefahrenzeichen*.

This leads us to the third layer, in which the meta-model is being defined. For *Gebotszeichen* the meta-model characteristics are its rectangular shape, its blue color, and its property of describing a command (as e.g. "there is a pedestrian crossing ahead, pay attention when driving a car and give priority to pedestrians who aim to cross the road"). Similarly, *Verbotszeichen* are characterized by it circular shape, red line on the outer boarder, and its property of prohibiting a certain act, and *Gefahrenzeichen* by its triangular shape, red line on the outer boarder and its property of pointing out a danger for road users. In return, the meta-models inherit their attributes to the respective models as described in layer 2.

Layer 4, the meta^2 model layer, describes the purpose of traffic signs (independent of their shapes and colors). Actually, traffic signs are designed to point out traffic rules described in length (and sometimes also with literal complexity) in law texts such as e.g. in Switzerland in the *Strassenverkehrsgesetz—SVG* and in

[1] *Gebotszeichen*, *Verbotszeichen* and *Gefahrenzeichen* are German words and classify traffic signs. They can be translated to English as "mandatory signs", "prohibition signs", and "danger signs".

Germany and Austria in their respective *Straßenverkehrsordnung - StVO*.[2] However, it is hard to imagine that car drivers know all relevant law texts by heart while driving along a road. This is why traffic signs help and remind car drivers of the traffic rules (as e.g. how to act in front of a pedestrian crossing in case a pedestrian indicates the wish to cross a road). Hence, the meta^{2}model containing the traffic rules is instantiated to meta-models of the three types of traffic signs, which again are instantiated into models of all varieties of the three types of traffic signs, which finally are instantiated to real-life traffic signs installed by the roadside.

How does a *stop sign* (as we see it in Central Europe) or a sign indicating *give priority* or a sign confirming to the car driver that he is driving on a priority road fit into this model? The stop sign has got an octagonal shape, the priority sign is a triangle with its tip on the bottom and the priority road is indicated with a sign in the shape of a diamond. All three signs are examples for the power of the meta-modelling concept as these three signs are customizations of the existing meta-model layer for traffic rules as described above. In the old times, some stop signs had the characteristics of a *Verbotszeichen*, however the property of it circular shape has been changed to an octagonal one for allowing to identify the stop sign also from the back (i.e. without seeing what is printed on the sign's front). This seems to be an important customization for further increasing traffic safety. Equally, the *give priority* and *priority road* signs have been customized for their shapes. These customizations do not jeopardize the corresponding meta-modelling concept of traffic rules as only individual properties have been modified. The meta-modelling concept is powerful enough to allow for these modifications.

The example of traffic signs and traffic rules was chosen to describe the meta-modelling concept and its underlying meta-modelling language. One could apply many other use cases as e.g. types of houses or types of transportation. Hence, the meta-modelling concept can be universally applied for almost any domain.

In the next section, we get a bit more scientific again and look at how the concept of meta-modelling can be applied for Intellectual Capital Management and Reporting.

3 Applying the Concept of Meta-Modelling on Intellectual Capital Management and Reporting

Intellectual Capital is a term used in the economy describing companies (sometimes also referred to as *"knowledge companies"* [10]) that aim to gain a (major) competitive advantage by using their intellectual capital leading to a diversification of goods or services with the ambition to gain higher margins and/or profits [3, 4]. Intellectual Capital itself has been defined in diverse ways, the one that appears to me the most cohesive one is *"intellectual capital is intellectual material that has*

[2] SVG and StVO are laws regulating traffic in the mentioned countries.

been formalized, captured and leveraged to produce a higher-valued asset" [11]. [3 , p. 50] provides an overview of definitions of Intellectual Capital.

Intellectual Capital Management and its *Reporting* have been researched in the 1990s and 2000s by both scientists and practitioners. As a result, over the years several institutions and companies have published their own intellectual capital reports that are not necessarily compatible with each other. Hence, when reading the intellectual capital report of company A and the one published by company B, one can hardly compare their wealth of intellectual capital. Eventually, the information on intellectual capital is restricted to the company that published its own report.

The concept of meta-modelling offers—with its four layers—a possibility to compare diverse intellectual capital reports from different publishers, such as institutions or companies. In analogy to the use case of traffic rules and corresponding traffic signs, one can apply the meta-modelling concept to achieve comparability and expressiveness of intellectual capital reports (and consequently its management) [3, 12]. Many intellectual capital reports apply some analogue or at least similar indicators, which allow for their transformation into *meta-indicator* (also referred to as benchmarks). If one combines all meta-indicators of the desired number of intellectual capital reports, one can generate a meta-model of all corresponding intellectual capital reports (refers to layer 3 of the meta-modelling concept, see Sect. 2 of this book chapter). Eventually, one can instantiate from layer 3 to layer 2 (the model layer) to craft a model of a specific intellectual report with its indicators, which in turn can be instantiated to an edition of an intellectual capital report. Figure 3 depicts this procedure.

The corresponding meta-model for Intellectual Capital Management and Reporting is depicted in Fig. 4.

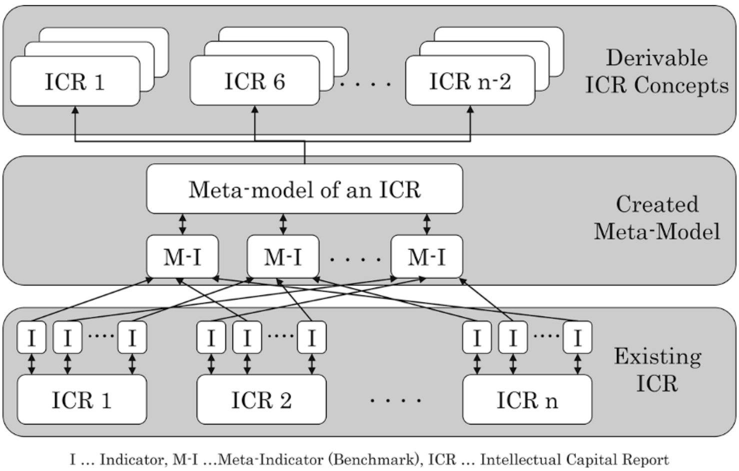


Fig. 3 Deriving intellectual capital reports from a meta-model [12]

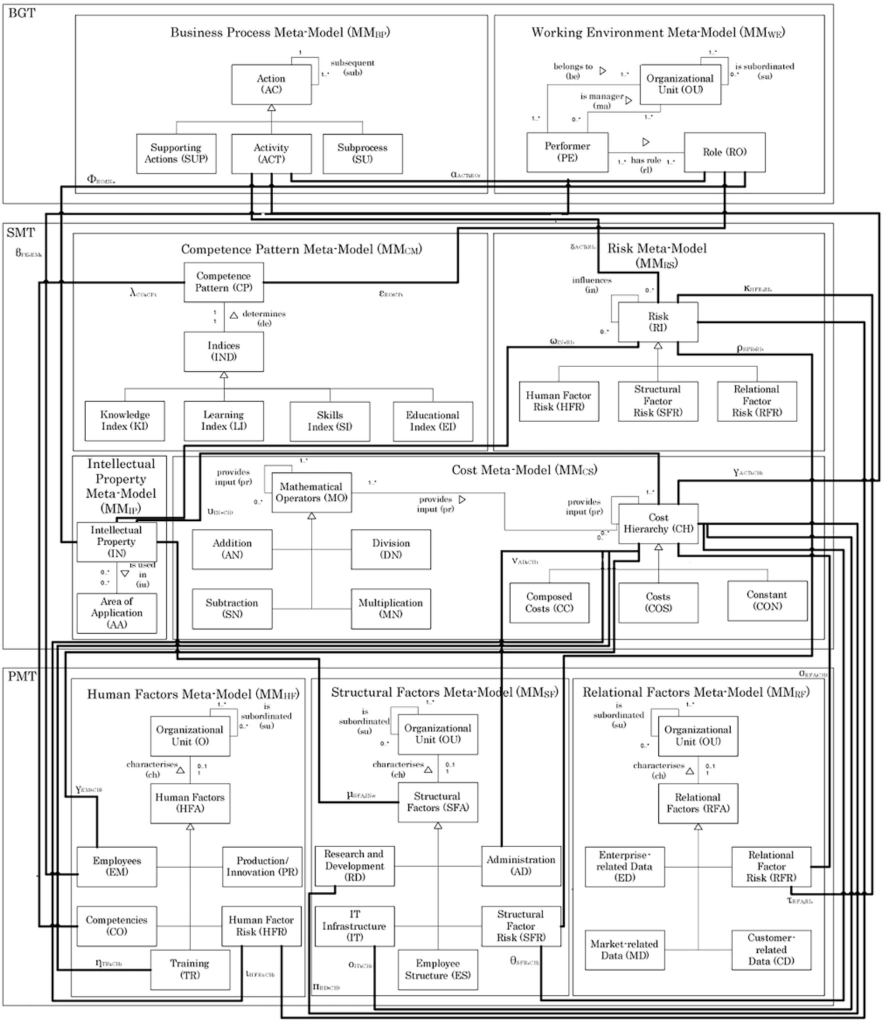


Fig. 4 Meta-model of the Intellectual Capital Management and Reporting framework [3]

Layers 2 and 3 (models and meta-models, respectively) have been implemented on the *ADOxx platform* [13] allowing for a computer-assisted transformation of companies' and/or institutions' individual intellectual capital reports. Hence, it is now possible to compare originally different reports with each other and enable the reader to interpret the amount, percentage, and value of intellectual capital within a certain company or institution and compare it with a report of another organization.

Although research in the field of intellectual capital has decreased since the financial crises in 2008 and 2009, the topic of intellectual capital (sometimes also referred to as *intangibles*) is more prominent than ever when looking at companies that drive digitalization and hence making tangible process steps increasingly intangible. The most obvious examples are Apple, Meta, Google, OpenAI and Microsoft but there are also more industry-specific (and less known) ones as e.g. Fieldwire [14] in the construction industry.

4 Examples of Applying Meta-Modelling in the Industry

As it has been described in Sect. 2, the concept of meta-modelling typically needs explanation for enabling users to apply it. *OMIlab* is a non-profit organization that is providing an environment, in which both scientists and practitioners can work with (meta-)models independent of their application in diverse domains [15]. We have initiated a cooperation with OMIlab for exploring the applicability of (meta-) modelling concepts in the definition and optimization of software-related processes [4].

4.1 OMIlab Workshops to Forecast Future Demands in Customer Support Setup

In September 2023, we have run a joint workshop for modelling support processes before implementing a software-based product allowing to track construction tools and whether they are loaded in a specific construction van [16]. The product is a combination of hardware, network, and software components. As the product is installed in a van, it is geographically mobile. This combination may lead to a complex situation once it comes to customer and product support in case the software-based product is not working as expected. For ensuring that customers experience a professional support service as well as a swift resolution of a specific issue that may have its root cause in one or more product parts (hardware, network and/or software), it is essential to find ways how to plan for relevant paths in issue resolution before the market launch of the product.

For this, we have applied the *Scene2Model* method [17] allowing for placing little paper figures on a canvas for describing a specific use case without needing to know the concept of (meta-)modelling. These paper figures have printed QR codes on them, which are scanned by a camera and automatically transferred as classes into the modelling editor of the meta-modelling platform *ADOxx* [13]. In the modelling editor the imported classes can be enriched with attributes (such as e.g. execution and waiting times or incurred costs). Eventually, the imported classes can be converted into classes of a specific modelling method such as BPMN. This enables analyses and simulation work, which is well-known from business process (model) optimization. Figure 5 depicts this transformation.

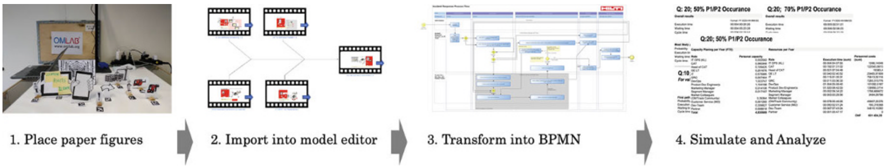


Fig. 5 Transformation from Scene2Model

As a result, different scenarios for a customer support setup could be formed including foreseen demands for staffing and expected costs. One of the variables for these scenarios was the estimated number for occurrences of highly severe and severe issues. Eventually, it enabled us to determine service level agreements per scenario.

The Scene2Model method with its paper figures was an intuitive starting point in this workshop as most participants have not been exposed to models or meta-models, respectively. The participants' feedback was encouraging so that in the meantime two additional workshops have been executed.

4.2 Complexity of IT and Software Solutions and Resulting Limitations of Meta-Modelling Tools

Most bigger companies run several complex IT and software solutions that are spanning across diverse technologies and that are being maintained and further developed by members from different teams.

Scientific literature (e.g. [18, 19]) as well as industry best-practices (e.g. [20]) emphasize the importance of detailed documentation and even (automatically generated) models of the solutions' architectures. Many companies suffer from insufficient documentation and precise architecture models due to different techniques and standards that are being applied. Oftentimes, architecture models are being *painted* without applying a common notation, e.g. in Microsoft Powerpoint, Microsoft Visio or other applications and stored as pictures in content management tools such as Confluence. Additionally, these architecture models (and their documentation) are scattered around different directories in content management tools, making it difficult to comprehend the complete architecture documentation of a large and complex IT or software solution. With the increasing degree of integration of today's IT and software solutions this issue gets more prevalent over time.

Some companies state that classic meta-modelling tools focusing on architecture documentation and modelling are not meeting their demand. Reasons being (amongst others) a lack of understanding of modelling and meta-modelling concepts (see Sect. 2) by users, complex tools with less intuitive user interfaces and finally a lack of integration of meta-modelling tools and their methods into the working practices of architects and users.

A way forward could lie in applying methods such as Scene2Model (see above) for architecture modelling. Another approach could be seen in the application of Artificial Intelligence (AI) tools.

5 Outlook: How Could AI Support in Documenting Complex IT and Software Solutions

As described above many companies have stored pictures of (segmented) architecture models in their content management systems. These pictures can serve as a basis for the AI-supported creation of IT or software architecture models that can be enriched with attributes. While AI tools can support in the creation of coherent architecture models stemming from individual pictures, software architects are needed to supervise, correct, and enrich the created models. However, this approach may provide a first step in getting towards a comprehensive architecture model for a complex IT and software solution. Combining the AI-supported creation of architecture models with the Scene2Model approach could lead to the creation of models and documentation that encompass complete IT and software solutions. Based on these models, software enhancements (e.g. features, enablers) could be planned in a more structured way. Additionally, simulations of how the complex IT or software solution would behave when adding a feature or an enabler could be run. This may lead to decreased costs for software architecture design and implementation with less efforts for bug resolution or—in worst case—refactoring work. First scientific papers have been published in this field (as e.g. [21]). However, for realizing this ambition, more research work is needed, at best in partnership between scientists and practitioners.

Acknowledgements I thank my colleagues Prof. Dr. Hans-Georg Fill and Dr. Harald Kuehn for inviting me to contribute to this *liber amicorum* honoring o.Univ.-Prof. Prof. h.c. Dr. Dimitris Karagiannis and to enable me to share my experience from past research and current work in industry.

I wish to specifically thank my doctoral advisor and mentor, Prof. Karagiannis for his year-long support and our many interesting and eye-opening discussions. In German language, we call the doctor advisor *Doktorvater* (literally means doctoral father). I am grateful to call Prof. Karagiannis my *Doktorvater*.

References

1. Karagiannis, D.: Business Process Management. University Course Scriptum, Winter term 2002/2003, University of Vienna
2. Hevner, A.R., Salvatore, T.M., Jinsoo, P., Sudha, R.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
3. Nemetz, M.: Intellectual Capital Management and Reporting – Achieving Comparability and Expressiveness by Applying the Meta-Modelling Approach. Suedwestdeutscher Verlag fuer Hochschulschriften, Saarbruecken (2009)

4. Karagiannis, D., Kuehn, H.: Meta-modelling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) Proceedings of the Third International Conference on EC- Web 2002, pp 182-95. Springer, Berlin (2002)
5. Karagiannis, D.: Metamodeling as a concept. In: Haring, G., Karagiannis, D. (eds.) Festschrift – Informationssysteme: Daten-Information-Wissen, pp. 83–99. OCG, Vienna (2007)
6. Karagiannis, D., Fill, H.-G., Hoefferer, P., Nemetz, M.: Metamodeling: Some application areas in information systems. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) Information Systems and e-Business Technologies. UNISCON 2008. Lecture Notes in Business Information Processing, vol. 5, pp. 175–188. Springer, Berlin, Heidelberg (2008)
7. Strahringer, S.: Metamodellierung als Instrument des Methodenvergleichs. Shaker Verlag, Aachen (1996)
8. Kuehn, H., Junginger, S., Karagiannis, D., Petersen, C.: Meta-Modellierung im Geschaeftsprozessmanagement. In: Desel, J., Pohr, K., Schuerr, A. (eds.) Modellierung '99, pp. 75–90. Teubner Verlag, Stuttgart (1999)
9. Hoefferer, P.: Achieving business process model interoperability using meta-models and ontologies. In: Oesterle, H., Schelp, J., Winter, R. (eds), Proceedings of the 15th European Conference on Information Systems (ECIS 2007), pp. 1620–1631, St. Gallen (2007)
10. Sullivan, P.H.: Basic definitions and concepts. In: Sullivan, P. (ed.) Profiting from Intellectual Capital, pp. 19–34. Wileyss, New York (1998)
11. Prusak, L.: Bibliography. http://www.laurenceprusak.com (2006) Accessed 27 Apr 2006
12. Nemetz, M.: Towards a model for creating comparable intellectual capital reports. Js Univ Knowl Manag. **1**(3), 217–234 (2006)
13. OMIlab: ADOxx Modules. http://www.omilab.org/adoxx/modules (2023) Accessed on 20 Dec 2023
14. Fieldwire: JobSite Management for Construction Teams. http://www.fieldwire.com (2023) Accessed on 20 Dec 2023
15. OMIlab: Our Philosophy. http://www.omilab.org/about (2023) Accessed on 20 Dec 2023
16. Hilti: Track your Vehicles and Tools with our Asset Tracking Solution. https://www.hilti.co.uk/content/hilti/E1/GB/en/business/business/equipment/on-track/van_gateway.html (2023) Accessed on 20 Dec 2023
17. OMIlab: Design Thinking. http://www.omilab.org/nodes/design-thinking (2023) Accessed on 20 Dec 2023
18. Lankhorst, M.: Enterprise architecture at work. Springer, Berlin, Heidelberg (2012)
19. Shanks, G., Gloet, M., Someh, I.A., Frampton, K., Tamm, T.: Achieving benefits with enterprise architecture. J. Strateg. Inf. Syst. **27**(2), 139–156 (2018)
20. BOC Group: Leverage Enterprise Architecture to Outpace Your Competition. http://www.boc-group.com/en/adoit (2023) Accessed on 20 Dec 2023
21. Heyn, H.-M., Knauss, E., Pelliccione, P.: A compositional approach to creating architecture frameworks with an application to distributed AI systems. J. Syst. Softw. 198 (April 2023)

From Static Models to Dynamic Systems: The Evolution of Business Process Simulation to Digital Twins

Christoph Prackwieser

Abstract The paper explores the progression from traditional process simulation techniques to the advanced concept of Digital Twins within the context of business processes and supply chain management. Initially, it outlines various discrete-event simulations and their suitability for analyzing business processes. Furthermore, it introduces the author's developments in simulation algorithms, particularly the "Hybrid Model Simulation" and "SIMchronization" methods. The Hybrid Model Simulation approach allows simulation across different control flow-oriented modeling notations. In contrast, SIMchronization focuses on synchronizing material and information flows in supply chain networks. A significant focus is on Digital Twins that integrate models, simulations, and real-time data to provide a dynamic representation of physical systems. The work culminates in proposing an extended SIMchronization method, integrating real-time data interfaces and artificial intelligence (AI) models to provide dynamic representation and adaptive behavior modeling of physical systems. This proposed extension aims to bring the capabilities of Digital Twins to the SIMchronization method, enhancing its utility in rapidly changing real-world environments. Overall, the paper contributes to the understanding of the evolution and application of simulation techniques in business process management, highlighting the transformative impact of Digital Twins and the potential of integrating AI and real-time data into existing simulation methodologies.

Keywords Simulation · Supply chain · Digital twin

C. Prackwieser (✉)
Adoplus Consulting Pty Ltd, Erskineville, NSW, Australia
e-mail: chris@adoplus.com.au

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_11

1 Introduction

The main goal of modeling and subsequent simulation is to replicate the behavior of dynamic systems and predict their behavior, required inputs, and produced outputs with reasonable effort. This paper focuses on business processes and supply chains, which are dynamic systems set in organizations and across business networks. In the literature, there are numerous definitions of the term simulation; Frank [1] provides an in-depth discussion on the concept of simulation. For this paper, particularly relevant is that simulation [2]

- is conducted based on a model of reality
- is computer-based and repeatable
- depicts the behavior and state changes of the model over time and provides the results
- allows for the alteration of input parameters and thus experimenting with scenarios
- delivers comprehensible results for the application domain, leading to a better understanding of the interrelations and thus identifying and evaluating potential improvements.

In the broader domain of supply chain management, simulation is used to support strategic decision-making and improve the operation [3]. The following simulation techniques are common [4]:

- Simulation in a spreadsheet software
- System Dynamics (SD)
- Discrete-event simulation
- Business Games

Discrete-event simulation is a suitable analysis tool for business processes. It can be very difficult to predict a system's response over time with a mathematical analytic model [5, 6] even for relatively simple systems. It becomes yet more difficult with an increase in relations within a network, for example, by adding interaction possibilities between processing stations or partners as in a typical supply chain.

A disadvantage of simulation is that it doesn't deliver an optimal solution directly and requires an experienced user to achieve a usable result. A further point of critique is that building simulation models and providing input data is a specialist's job [7] and takes some time. Therefore, changes in the real world may not get reflected in the model fast enough, and variations in the input data will not influence the simulation as there is no real-time data feed from the simulated environment to the simulation model.

In 2003, Grieves introduced the concept of Digital Twins, that incorporates models, simulation, and real-time data in a total product lifecycle management course. Since then, the use cases, definition, and maturity of Digital Twins have constantly evolved.

Already 8 years earlier, in 1995, Karagiannis [8] introduced the Business Process Management Systems (BPMS) methodology framework. This concept that can be seen as a precursor to a modern Digital Twin. BPMS combines virtual models of executable business processes and organizational structures, a re-engineering component that is supported by simulation, animation, and analytics, and a "Performance Evaluation Process" that incorporates performance data from the analyst's "real-world" system. The author remarks that the vast amount of data from executing a BPMS application is invaluable input for strategic decision-making and operational improvements.

1.1 Overview of Simulation Algorithms

The simulation algorithms discussed in the following chapters represent a sequence of maturity and capability uplift steps in business process simulation. The author of this paper was involved developing a range of discrete-event simulation algorithms and tools for both business processes and supply chains. The discussed tools are either functionalities of the commercial product ADONIS® or implemented as add-ons on the metamodeling platform ADOxx® [19]. (ADONIS® and ADOxx® are registered trademarks of BOC Products & Services AG).

Path Analysis. This algorithm tries to find all possible pathways through a business process model and calculate each path's likelihood, execution duration, cost, and other criteria. It is applied to an individual process model only, including its subprocesses.

Capacity Analysis. This simulation algorithm analyses processes in conjunction with the resources available to execute the processes' activities. In the so-called "ADONIS® Standard Modeling Notation", the process flows and resources are modeled in different modeling notations and are created in separate instances of their respective model type. Each activity in a process is linked to a resource. The simulation evaluates those relationships at runtime and can determine, for example, the capacity requirements per role, the average cost per process execution, and the cost caused per cost center.

Workload Analysis. Uses the same models as the Capacity Analysis and adds calendars of stochastic process start times and resource availability. The algorithm is based on a waiting queue model that automatically determines waiting times at activities with occupied performers. Consequently, it helps with analyzing lead times and identifying bottlenecks in the process.

When the author developed the "Hybrid Model Simulation" approach presented below, the simulation algorithm of ADONIS®, like many Business Process Management Tools, was optimized for its modeling notation. Even though the underlying metamodel provided some flexibility, it was not enough to simulate processes designed in other control-flow-based notations, such as EPC or BPMN. Since then, ADONIS® has gained the capability to simulate BPMN models natively.

To overcome this initial limitation, the author developed the "Hybrid Model Simulation" approach that can simulate and animate processes of varied notations, even if they are part of the same overarching process model.

2 Hybrid Model Simulation

A primary motivation for developing this approach [9] was the "Hybrid Method Engineering" approach introduced by Karagiannis and Visic [10] as part of the "Next Generation Modeling Framework" (NGMF).

This algorithm is unique in its ability to simulate and animate process models across various control flow-oriented modeling notations. The basic idea of this approach is to leave the graphical representation of each model unchanged for the simulation run to improve the user's identification with and understanding of the simulation results. The generic simulation "Hybrid Model Simulation" algorithm can be applied on any control flow-oriented process model (Fig. 1).

A list of semantic simulation core concepts was compiled to map a wide range of classes of control-flow-oriented modeling languages. A flow rule language was introduced to translate the execution sequence of a model that is expressed with arrows or events. Before a simulation run, the internal semantic mapping module automatically identifies the suitable core concept, creates the sequencing rule, and annotates both to the modeling object.

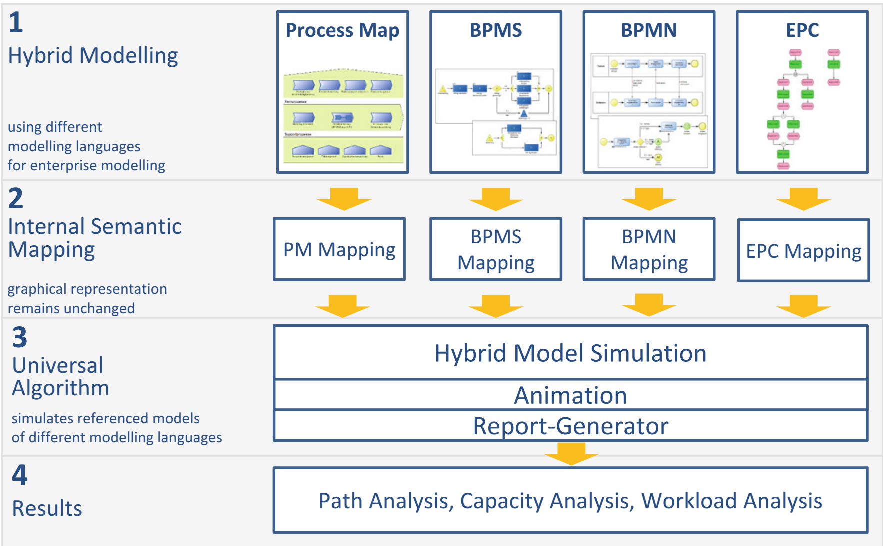
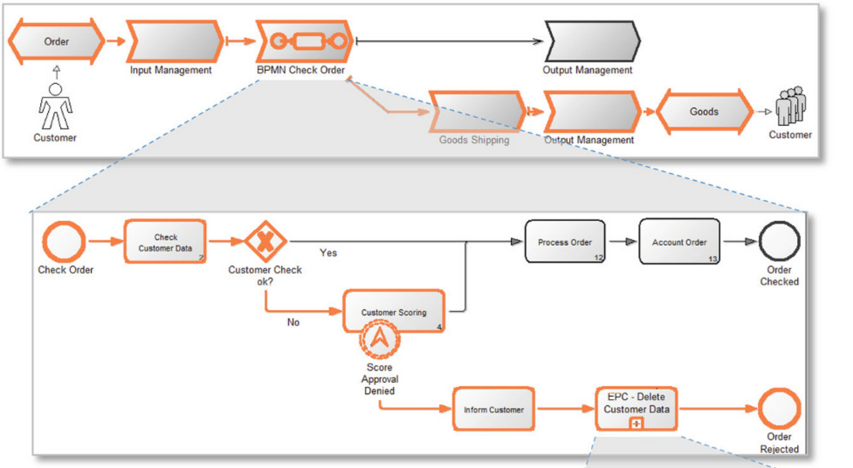


Fig. 1 Hybrid Model Simulation—Concept



Path	Prob-ability	Execution Time	Wating Time	Cycle Time	Path Description: Object	Type	Model
Path 1	40.00%	64	0	64			
Path 2	10.00%	18	0	18	Order	Input	VC Order Processing
					Input Management	Process	VC Order Processing
					BPMN Check Order	Process	VC Order Processing
					Check Order	Start Event	BPMN Check Order
					Check Cust. Data	Task	BPMN Check Order
					Cust. Check ok?	Excl. OR	BPMN Check Order
					Cust. Scoring	Task	BPMN Check Order
					Score Appr. Denied	Interm. Eve	BPMN Check Order
					Inform Cust.	Task	BPMN Check Order
					Delete Cust. Data	Sub Process	BPMN Check Order
					Delete Cust. Data	Event	EPC Delete Cust. Data
					Find Cust.	Function	EPC Delete Cust. Data
					XOR	XOR	EPC Delete Cust. Data
					Account existant	Event	EPC Delete Cust. Data
					Delete Cust. Account	Function	EPC Delete Cust. Data
					Account deleted	Event	EPC Delete Cust. Data
					XOR	XOR	EPC Delete Cust. Data
					Delete Cust. Data	Function	EPC Delete Cust. Data
					Cust. Data Deleted	Event	EPC Delete Cust. Data
					Order Rejected	End Event	BPMN Check Order
					Goods Shipping	Process	VC Order Processing
					Output Management	Process	VC Order Processing
					Goods	Output	VC Order Processing
Path 3	50.00%	70	0	70			
Sum	100.00%	152	0	152			

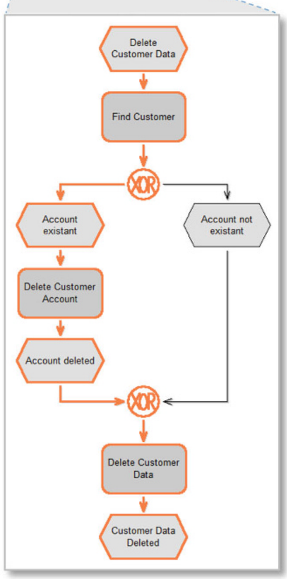


Fig. 2 Hybrid process model and simulation results [9]

The simulation has an animation component that displays tokens moving through the model to provide the user with an animated process that supports understanding (Fig. 2).

As stated before, this algorithm works for control-flow-oriented languages but not for supply chain models that incorporate information flows and material flows. To simulate those, the author developed a simulation method called "SIMchronization."

3 SIMchronization

SIMchronization [2, 11] is a method developed to enhance the productivity and responsiveness of supply chain networks. This approach emphasizes the crucial need for a tight and instantaneous coupling of information and material flows within these networks. Utilizing a unique combination of a domain-specific graphical modeling language and behavior-describing rule sets alongside a discrete simulation algorithm, SIMchronization effectively unveils the complex and highly dynamic interactions within supply chains. A key output of this method is the generation of 'State Flow Diagrams,' which visually represent the information and material flows, such as sent messages and processed parts, sequentially. These diagrams play a pivotal role in illustrating stock-level developments over time, aid in synchronizing supply chain processes, and making them valuable for communicating operational concepts for new or modified supply chains (Fig. 3).

3.1 Static Supply Chain Model

The foundation of SIMchronization lies in its ability to analyze and synchronize material and information flows within a supply chain. Initially conceived for e-maintenance supply chains, the method is adaptable to various logistic environments. Its analytical component involves a graphical model outlining the supply chain structure, augmented by behavior-describing rules for the elements

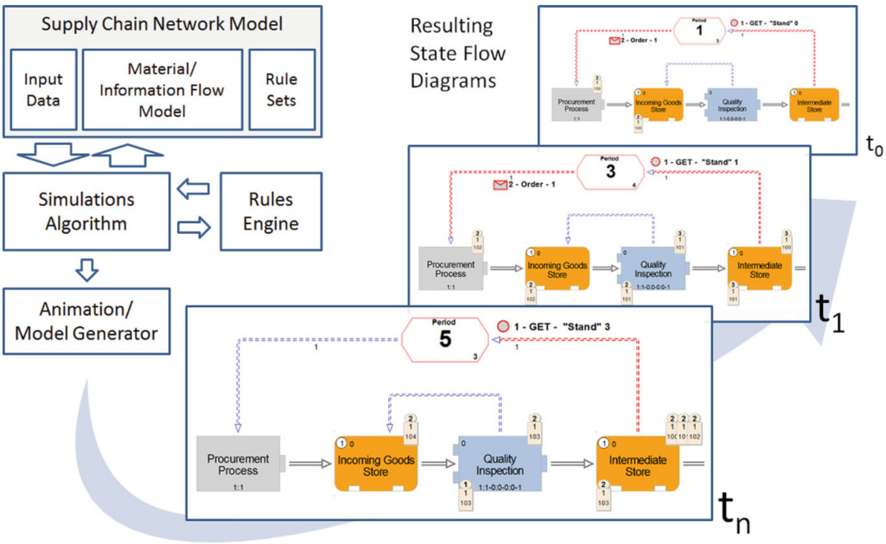


Fig. 3 The SIMchronization method [12]

and stakeholders involved. This model incorporates SCOR (Supply Chain Operations Reference Model) components, with modeling classes like ‘Source,’ ‘Store,’ ‘Make,’ ‘Transport,’ ‘Deliver,’ ‘Reader,’ and ‘Switch.’ The ‘Plan’ class is essential, encompassing production planning and centralized supply chain control processes. Material and information flow channels within a model facilitate the transfer of parts and communication between objects. The inclusion of Auto-ID technologies, such as RFID, OCR, or barcode systems, further enables the method to analyze autonomously controlled logistic processes effectively.

3.2 Rule Sets

In addressing the dynamic nature of supply chain interactions, especially under changing conditions, SIMchronization integrates formal rule sets that describe the behavior of supply chain elements in various scenarios. These rules go beyond what can be represented through a graphical model alone, offering commands that influence the information and material flows within the supply chain. For example, in a centralized control scenario, a ‘Plan’ process might monitor sales volumes from a ‘Deliver’ process and initiate corresponding orders to a ‘Source’ process. This intricate interplay of rules ensures the system can adapt and respond to diverse operational situations. Rules also define execution times, required input factors, and generated outputs of “Make” process steps along the supply chain.

Furthermore, an essential aspect of every maintenance policy is to decide if an asset is maintained proactively or reactively after a failure. A proactive maintenance regime can either define a fixed time between maintenance tasks or be based on the asset’s condition. SIMchronization uses rule sets to model the condition and deterioration of an asset over time. Additionally, rules can feed unexpected events, such as breakdowns or other external influences, into the supply chain.

3.3 Simulation

The dynamic aspects of supply chains are brought to light through a discrete simulation algorithm applied to the static model and the rule sets. This simulation process involves a priority and event sequence, where the rule set of an object is read periodically and processed by a rule engine. The engine evaluates the rules and feeds the results into the simulation, facilitating the creation of production orders and the movement of parts within the simulated environment. The simulation not only animates the flow of parts through the supply chain but also provides vital quantitative data post-simulation, such as lead times and activity-based costs. This data, combined with the generated state flow diagrams for each simulation period, enhances the understanding and communication of the operational aspects of newly designed or modified supply chains.

4 Digital Twins

Besides the Internet of Things, Artificial Intelligence, Blockchain, and 3D-Printing, Digital Twins are one of the newer enabling technologies of Industry 4.0 to implement smart supply chains and manufacturing. They can also be a step-change for organizations on their journey to accomplish industrial digital transformation.

A Digital Twin is a virtual, dynamic representation of a physical system [13]. It couples actual assets, networks, or systems in real-time with a corresponding virtual replica. It uses models, data, sensors, analytics, dashboards, and human-machine interfaces such as Augmented Reality (AR) and Virtual Reality (VR)-supported visualization to provide insights and operational and decision support [14]. The most advanced Digital Twins control production processes autonomously and develop maintenance plans and schedules.

A definition that is often cited in scientific papers is given by Glaessgen and Stargel [15]. They see a Digital Twin as an integrated multi-physics and multi-scale, probabilistic simulation of a complex product that utilizes the best available technologies, such as physical models and sensors, to mirror the life of its corresponding twin.

Working with the digital replicate in the virtual space allows designers and operators of machinery and systems to test out alternative scenarios while receiving real-time feedback from the Digital Twin. Engineers can virtually test improvements and designs before implementation so that initial problems can be detected and solved quickly. They can simulate different scenarios and assess and compare them against each other. This uplift in the agility of the development and operations process leads to quicker decision-making, reduced cost, and better designs and products. Early Digital Twins were developed to support the maintenance of assets, especially to predict and detect issues before they lead to failures. They allow for a proactive or predictive maintenance regime that reduces downtime of assets and costs and improves safety and quality.

Early adaptors of this technology can be found in the domain of aircraft maintenance as, for example, aircraft engines are very expensive assets that need to be highly reliable [16]. Furthermore, engines have numerous sensors built in and provide the necessary data stream to update the virtual twin in real-time. From a quality assurance and maintenance perspective, it is especially interesting that engine manufacturers can compare data they receive from their whole fleet of deployed engines. Each engine may have its own Digital Twin that mirrors its condition and performance. This allows the service team to learn from the behavior of the whole fleet by training a machine-learning model with the data and inspection reports received. Each individual Digital Twin can then use the trained machine learning model to detect anomalies in its assigned real-world engine, predict failure times, raise alarms, or create maintenance work orders.

The main components of a digital twin are:

- Physical asset(s) or system(s): the real-world object or objects the digital twin represents.

- Digital representation: a detailed, dynamic, and virtual model of the physical asset. The digital representation could be, for example:
  - a physical model to analyze structural integrity or thermodynamics,
  - a 3D model of an asset or building,
  - a process model that depicts the workflows and operations of a system, such as a supply chain model in SIMchronization or
  - a behavioral model that models the system's reactions under various conditions and inputs.

  The digital representation can also be a combination of different types of models.
- Connectivity and data transmission: the data link between the physical asset and its digital representation. The data could be measurements of the condition of an asset collected by IoT sensors. In advanced Digital Twins, the data flow can be bi-directional, enabling monitoring and control arrangement.
- Data analytics: the collected data is processed to derive insights and potentially recommend decisions. Simulation and advanced analytics, such as machine learning, are used to identify trends, find anomalies, generate predictions, and improve performance. The data will be analyzed considering the digital representation, such as network graphs for transport networks.
- User Interface: depending on the application and user group, a range of user interfaces can provide intuitive access to the data itself or derived insights. For example, dashboards, animations, 3D model visualizations, Virtual Reality, and Augmented Reality are widely used Digital Twin interfaces.
- Control and feedback loops: Advanced Digital Twins may provide insights and information to the physical asset. Some send control commands to the asset or system. If the Digital Twin identifies issues or a need for action, it may create a work order or raise an alarm.
- Security and Privacy: as with every connected digital system, secure design and operation of the system is of the highest priority to protect the data, physical assets, and users from a cyber-attack.

5 The Extended SIMchronization Method

5.1 Comparison of Digital Twins and SIMchronization

This chapter explores how to extend the standard SIMchronization method to a Digital Twin. The author proposes an extension of the method with concepts that enable the user of SIMchronization to benefit from capabilities that are so far known only to Digital Twins. Firstly, the two approaches are compared, and the missing components in the SIMchronization method are identified:

Physical assets or system. The initial SIMchronization method [2] was developed for the domain of (e-)maintenance supply chains. Therefore, the real-world objects that the method aims to analyze and improve are:

- all activities required for planning, managing, and performing a maintenance process and their interactions, such as material flows and information flows and
- the assets that condition need to be monitored and maintained over time.

Digital representation. SIMchronization uses Supply Chain Network Models to depict the static relationships and rules to define the behavior, status, and reactions to changing inputs and conditions.

Connectivity and data transmission. SIMchronization cannot ingest data in real time. Data required for simulation and analysis is acquired before the simulation run and stored in a static fact base that is part of the model.

*Data Analytics. S*IMchronization incorporates a discrete simulation and animation component that is applied to dynamize the model. The user gains insights by studying simulation results and observing the animation to understand better the model's interdependencies, impacts of improvements, and the supply chain's resilience towards external events.

User Interface. To evaluate the initial approach, a prototype was implemented by using the meta-modeling platform ADOxx®. The modeling classes and their attributes were created within the software, and the simulation and animation algorithm, combined with the rules interpreter, were coded in the scripting language (AdoScript) of the platform. The animation provides a two-dimensional view of the model.

Control and feedback loops. SIMchronization does not provide data or information to the real-world asset, just to the user. A bidirectional data flow may be a concept for a future extension of the method.

Security and Privacy. As the initial method is not connected to real-world objects or the internet, the implemented prototype relies only on the security capability of the underlying platform.

One of the main differences between the two approaches is the access to real-time performance data. While the entanglement of real and virtual objects is crucial for a Digital Twin, the initial SIMchronization method relies on the provision of historical data.

Consequently, to enable SIMchronization to quicker react to changes in the real world and to provide ad-hoc decision support the author proposes to add a live data feed to the method. As unforeseen events may occur and real-time data can include unexpected data points, the pre-defined set of rules may not cover all eventualities. Therefore, a more flexible, potentially continuously developing behavioral model is required. For this, we propose an AI model that predicts the condition or behavior of an entity or system under a specific set of input parameters, see Fig. 4.

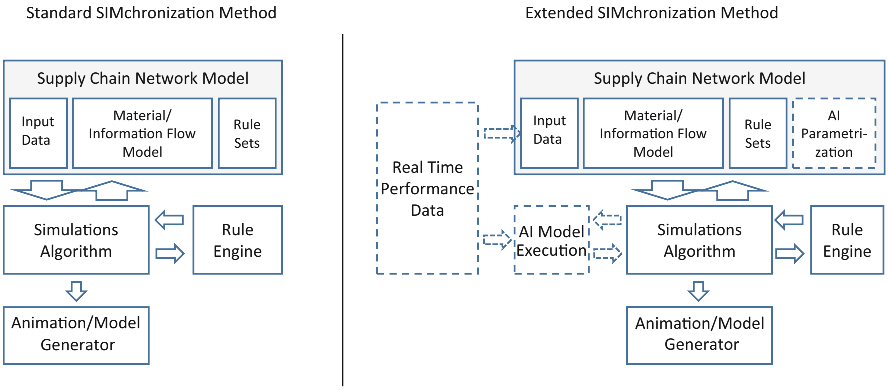


Fig. 4 Extended SIMchronization Method – Change to AI Model and Parametrization

5.2 Real-Time Performance Data

To achieve a dynamic representation of a physical system, Digital Twins need to be updated with real-time data of the performance and condition of the represented system. For example, this data may come via the Internet of Things (IoT) directly from sensors attached to assets that need to be maintained or could come from Inventory Management Systems reporting current stock levels.

In its initial design, SIMchronization sources the input data it uses in the simulation from a fact base. The fact base is a data store filled with historical data entered into the model before a simulation run starts.

The extended SIMchronization method uses its connectors to update the fact base in real-time and provide live and historical data as input into the simulation algorithm, the connected rules engine, and the AI prediction models.

5.3 Artificial Intelligence Model

The AI model can be seen as a complementary module to the Rule Engine that provides the required adaptability and scalability. This paper does not propose a specific AI method; depending on the domain, SIMchronization is applied with different AI techniques that may be suitable [17]. In the maintenance space, for example, an algorithm that excels in forecasting an asset's condition and one that can predict the system's reaction to specific input factors is required. Typical advantages of AI technologies that can be beneficial for SIMchronization's application are:

- Forecasting/prediction capability
- Anomaly detection and pattern recognition

- Feature extraction
- Handling of non-linearity
- Scalability and handling of multiple data sources
- Adaptability

A disadvantage of using some more advanced AI/ML algorithms is the lack of interpretability, especially when the understanding of the underlying model decisions is crucial.

5.4 AI Parametrization

Parametrization [18] of an AI model refers to the process of defining and adjusting the parameters that control the behavior and output of an artificial intelligence system. The number of parameters can be large depending on the type of the AI model and its intended application. An initial parametrization takes place during the learning process. In the case of a neural network, the parameters are the structural settings of the network and weights and biases. The model's performance largely depends on the proper selection of these parameters.

A runtime parametrization can adjust and fine-tune the model's parameters in response to new data or changing conditions in real-time or near-real-time to enable the dynamic adaptation of the AI model to changes in the environment and input data during its operational execution. This continuous learning approach increases the model's flexibility and relevance in dynamic environments but also requires careful management to maintain stability and accuracy and to prevent overfitting.

6 Summary

This paper presented a range of discrete-event simulation algorithms and methods that can be applied to control-flow-based modeling notations such as business process and supply chain network models. The models range from a simple Path Analysis that analyses a single process model to a Capacity and Workload Analysis that simulates models of two model types, at least one process, and a resource model, in conjunction. The Hybrid Model Simulation approach can simulate models of any control-flow-oriented notation together in one run. Finally, the SIMchronization method doesn't use a control flow but simulates material and information flows through a supply chain network.

Digital Twins use models, simulations, and real-time data to mirror real-world objects in virtual space. There are advantages to having a realistic digital representation of an asset or system in which scenarios can be run and improvements planned.

To provide similar capabilities to the SIMchronization method, the author proposes a real-time data interface and an artificial model to support system behavior and asset status modeling.

Some suggestions in the literature exist on which AI technology would suit this task [17]. However, more research is required. Another interesting research field and a potential extension of the SIMchronization method would be to use real-time data from the system to feed the fact base of the simulation and to use generative AI to create and change the static model itself.

References

1. Biethahn, J., Hummeltenberg, W., Schmidt, B., Stähly, P., Witte, T.: Simulation als betriebliche Entscheidungshilfe: State of the Art und neuere Entwicklungen. Springer (2013)
2. Prackwieser, C.S.: Simulation von Material- und Informationsflüssen zur Analyse und Verbesserung von Instandhaltungsprozessen, Vienna (2017)
3. Kleijnen, J.P.C.: Supply chain simulation tools and techniques: a survey. Int. J. Simul. Process. Model. **1**, 82–89 (2005)
4. Kleijnen, J.P.C., Smits, M.T.: Performance metrics in supply chain management. J. Oper. Res. Soc. **54**, 507–514 (2003)
5. Bossel, H.: Systeme, Dynamik, Simulation: Modell bildung, Analyse und Simulation komplexer Systeme. BoD–Books on Demand (2004)
6. Huang, G.Q., Lau, J.S.K., Mak, K.L.: The impacts of sharing production information on supply chain dynamics: A review of the literature. Int. J. Prod. Res. **41**, 1483–1517 (2003)
7. Banks, J.: Introduction to simulation. WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation – A Bridge to the Future' (Cat. No.99CH37038) (1999)
8. Karagiannis, D.: BPMS: business process management systems. SIGOIS Bull. **16**, 10–13 (1995)
9. Prackwieser, C., Buchmann, R.A., Grossmann, W., Karagiannis, D.: Towards a Generic Hybrid Simulation Algorithm Based on a Semantic Mapping and Rule Evaluation Approach. The 2013 International Conference on Knowledge Science, Engineering and Management, pp. 147–160, Dalian (2013)
10. Karagiannis, D., Visic, N.: Next generation of modelling platforms. In: Grabis, J., Kirikova, M. (eds.) Perspectives in Business Informatics Research, vol. 90, pp. 19–28. Springer, Berlin, Heidelberg (2011)
11. Prackwieser, C.: A modeling procedure for information and material flow analysis comprising graphical models, rules and animated simulation. In: Buchmann, R., Kifor, C.V., Yu, J. (eds.) Knowledge Science, Engineering and Management, Ksem 2014, vol. 8793, pp. 198–209. Springer, Berlin, Berlin (2014)
12. Prackwieser, C.: SIMchronization: a method supporting the synchronisation of information and material flows. Proceedings of the Winter Simulation Conference, pp. 1–2. Winter Simulation Conference, Berlin (2012)
13. Fu, Y., Zhu, G., Zhu, M., Xuan, F.: Digital twin for integration of design-manufacturing-maintenance: an overview. Chin. J. Mech. Eng. **35**, 80 (2022)
14. Attaran, M., Celik, B.G.: Digital Twin: benefits, use cases, challenges, and opportunities. Decis Anal J. **6**, 100165 (2023)
15. Glaessgen, E., Stargel, D.: The digital twin paradigm for future NASA and U.S. air force vehicles (2012)
16. Wu, Z., Li, J.: A framework of dynamic data driven digital twin for complex engineering products: the example of aircraft engine health management. Procedia Manuf. **55**, 139–146 (2021)

17. Emmert-Streib, F.: What is the role of AI for digital twins? AI. **4**, 721–728 (2023)
18. Huang, Z., Fey, M., Liu, C., Beysel, E., Xu, X., Brecher, C.: Hybrid learning-based digital twin for manufacturing process: modeling framework and implementation. Robot. Comput. Integr. Manuf. **82**, 102545 (2023)
19. OMILAB - ADOxx Metdamodelling Platform https://www.omilab.org/adoxx , last accessed 2023/12/22

Collaborative Modeling Method Prototyping Using Digital Design Thinking with Scene2Model

Wilfrid Utz

Abstract Realizing a modeling method is a knowledge intense task that requires the involvement of stakeholders from different backgrounds and expertise during all phases in the realization lifecycle. This lifecycle is typically coined "conceptualization" and spans from ideation, design, formalization, ~~and~~ implementation and deployment. The outcomes of design decisions taken as a team impact the capabilities provided by the modeling method from a structural and behavioral perspective. Multiple iterations are usually required to externalize the knowledge, agree on a formalization approach, and evaluate the design results through prototyping techniques. Following the Agile Modeling Method Engineering (AMME) practice proposed by Prof. Dimitris Karagiannis, this contribution introduces an instantiation in three parts of the engineering technique. First, it suggests haptic design interactions as an interface for the domain experts. Second, it derives an initial formalization of the metamodel skeleton through metamodel processing. Third, it enhances the design with functional/behavioral aspects and provides the outcome as a digital prototype for immediate testing/evaluation using the experimentation capabilities of the ADOxx metamodeling platform.

Keywords Metamodeling · Metamodel design operation · Design thinking · Scene2Model · ADOxx

1 Introduction and Motivation

The application of model-based approaches in information systems design has become a well-established technique to understand the domain requirements towards the system-under-study, capture those requirements in a structured way and utilize models as a means of documentation, communication, and nowadays also

W. Utz (✉)
OMiLAB NPO, Berlin, Germany
e-mail: wilfrid.utz@omilab.org

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_12

operational use of model artefacts. Operationalization requires that the modeling method is not only a theoretical and conceptual construct but tooling is provided to operate the modeling method and enable advanced processing functionalities. These functionalities add domain-specific value to the model as the artefacts created with the modeling method can be assessed, queried, enhanced, and transformed using its digital representation.

Considering the domain specificity and purpose as discussed in [1], it has become necessary to acknowledge the formalization of the syntax, notational aspects, and semantics as a means for the interaction of human experts and machines alike. Purpose is a central aspect of this consideration. Consequently, techniques are needed that support the engineer of a modeling method to efficiently externalize domain knowledge, create new constructs, and combine them with pre-existing fragments into an adequate artefact that satisfies the specific purposes of users.

The motivation of this contribution relates to the observation that conceptual modeling has evolved towards a commodity activity. The task is not exclusively related to experts in the field of knowledge engineering and management anymore, that have the capabilities to map domain knowledge into a formal representation, but to a broad viewpoint involving potentially any actor from an organization in the modeling task. Consequently, it is required that the needs of this set of actors become tangible during the conceptualization lifecycle.

Additionally, we can observe that the evolution of modeling methods happens at a higher pace than in the past because of digitalization and digital transformation needs. This means that temporal and agile evolution aspects in the design of the methods should be supported. Users expect that for their specific task, the required constructs evolve along the technological trends such as business ecosystem changes or compliance requirements. An integrated view is needed that has on a concrete level a shorter evolution lifecycle, but contributes individually, to the core concepts adequate for enterprise modeling in a specific organization.

Based on these observations, this paper proposes an approach that builds on the Agile Modeling Method Engineering (AMME) practice introduced in [2] and instantiates it with digital design thinking techniques. The instantiation consists of (a) a haptic design library for modeling methods using paper-based knowledge externalization, (b) collaborative interaction techniques, and (c) digitalization of the design into a digital twin based on the Concept-Characteristic-Connector (CoChaCo) [3] approach. This last element enables design processing functionalities and a language-based modeling tool prototype generator based on the open-source ADOxx [4] metamodeling platform.[1]

The remainder of this paper is structured in the following way: Sect. 2 introduces the related work that enables the proposed approach, specifically focusing on conceptual modeling, metamodeling, and design thinking. Section 3, as the core part, discusses the proposed design process and introduces the conceptual architecture required, focusing on user interaction and artefacts produced. For evaluation, a

[1] Open-source experimentation environment accessible via https://www.adoxx.org

prototyping approach has been applied, demonstrating the technique using a generic case from the work within the OMiLAB[2] Community of Practice, specifically the smart innovation environment for digital engineers [5] in Sect. 4. Conclusions and next steps are presented in Sect. 5.

2 Related Work

Considering the motivation of this contribution, it is relevant to investigate the literature on how modeling methods have been designed in the past, specifically in the field of domain-specific conceptual modeling.

Modeling Methods Specification Following the definition of the term "domain" as introduced in [1] in conceptual modeling, we can observe that general-purpose modeling languages such as BPMN [6] or UML [7] have a specific domain focus (process management for BPMN, software engineering for UML). As a result form standardization efforts and aiming to support a broad spectrum of application scenarios they typically consist of a large set of semantically related concepts and constructs. Consequently, their application is limited to experts in the field. When applying such a general-purpose modeling language, the challenge of the user relates mainly to identifying which construct is intended for what purpose and how to apply it properly. It therefore requires a detailed understanding of the modeling approach before utilization. Processing capabilities are limited as a result.

In contrast, domain-specific modeling approaches are highly specialized and establish a vocabulary that is distinct to the application domain. The motivation to realize a domain-specific language is attributed to a) the expert user interacting with it (expressiveness) and b) the processing of constructs (e.g. specific algorithmic implementation that requires model-based input such as simulation-capable models). An overview of implementation results can be found in [8, 9].

From the perspective of this contribution, both flavors of modeling methods are equally relevant. General-purpose languages are considered as commonly "agreed" knowledge for specific domain aspects. As such they are classified as foundational languages, and their specification is re-usable for standardized purposes. Domain-specific languages, in contrast, demonstrate the possibilities to enable specialized vocabularies and processing functionalities.

Modeling Method/Metamodel Representation In literature, it can be recognized that *technological capabilities* impact the way how modeling methods are *conceptualized*. Historically various approaches can be identified such as a language-based understanding in [10], an ontological viewpoint as discussed in [11, 12] or logic-based techniques introduced in [13, 14]. This means that when engineering a modeling method, the selected realization technology for constructs and/or required

[2] OMiLAB Community of Practice - Modeling Tools: https://www.omilab.org/activities/projects/

processing algorithms directly influences the design process. An example of this viewpoint can be found in [15] for BPMN. Even though the schema of the notation is specified in detail, various research articles can be found that showcase the need to represent the metamodel using ontology technologies such as OWL or RDF (need for advanced queries), logic, specifically TELOS to enable simulation or UML to allow for code generation of BPMN diagrams. The above observation results in re-conceptualizing with every new promising technology implementing metamodeling concepts [16].

From the viewpoint of this contribution, a representation is sought after (on a formal level) that can support any kind of representation and consequently processing functionality is discovered and aligned. The Concept-Characteristic-Connector (CoChaCo) approach as introduced in [3] is used for the prototypical implementation. As such the formalization requirement articulated in [17, 18] is supported on various levels.

Design Thinking for Modeling Method Engineering The design of a modeling method is a knowledge-intensive task and requires a detailed understanding of needs and requirements, capabilities of semantic technologies and expertise of the intended user. Following the classical definition of Brown in [19], design thinking is inspired by co-creation/collaboration in multi-disciplinary teams. This applies to the approach discussed in this paper as it is envisioned, that in the future modeling methods will evolve and be again decommissioned rapidly to enable a situational combination of various viewpoints of experts [20]. This requires a dynamic alignment of model-value functionality. Exploration in the form of prototypes is considered an important aspect during such design iterations.

3 Conceptual Approach: Design to Prototype

Based on practical observations in the context of the realization of domain-specific modeling methods within the OMiLAB Digital Innovation Environment, Fig. 1 depicts how modeling methods are typically conceptualized, implemented, and deployed.

Two acting roles are typically involved in the design and implementation process: the *modeler* as an expert in the domain observes the system-under-study, identifies the purpose of the conceptual model, and defines requirements for the modeling method using examples to explain the required capabilities. These requirements are then reflected by the *modeling method engineer*, the adequate metamodel technique is selected and applied to establish the structural and behavioral aspects of the modeling method. Design and realization operations are enabled based on the formalization embedded within the technique and support the modeling method engineer. The realized artefact is instantiated by the modeler to apply it to the scenario for testing purposes.

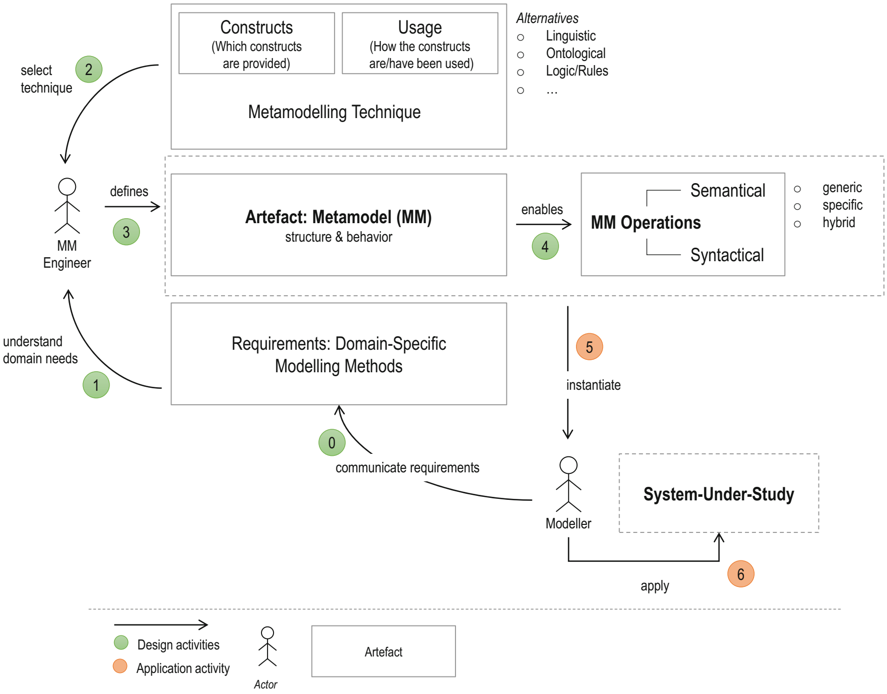


Fig. 1 Observation: modeling method engineering (based on [16])

Even though this approach is well structured, it encompasses certain limitations with respect to stakeholder involvement, agility, and iterative adaptation as well as prototyping support. These observations are listed below:

1. **Domain Knowledge Externalization.** The understanding of domain knowledge is distributed between the modeler and modeling method engineer in the traditional approach. This implies that a negotiation process is triggered for each requirement resulting in a design decision jointly taken.

   Considering a set of multi-disciplinary actors involved, the elicitation of requirements becomes challenging as design decisions need to be balanced and harmonized not only between a single expert user and the engineer but with a group of experts. Contradictory requirements stemming from different experts need to be supported as the user of the modeling method becomes its engineer.
2. **Concrete Metamodeling Techniques Impact Design.** Considering the definition of design thinking, freedom to design modeling methods is required, with as little as possible technical restrictions from the realization platform. Currently, we can observe that the expertise of the engineer impacts the design capabilities. Technological restrictions hinder and limit the design space.

   For the approach presented in this paper, we consider a design environment that abstracts upon existing metamodeling techniques. This results in a higher level of abstraction, reducing the functional expressiveness but enables arbitrary design possibilities. To move between the levels, design support operations are suggested.

3. **Design to Prototype.** Design thinking includes early testing of concepts through prototyping. Currently, the time-to-prototype strongly depends on the expertise of the modeling method engineer and the restrictions of the selected platform for operationalization.

   By utilizing a generator-based approach, is assumed that the impact of the design decision can be evaluated, at least to a certain extent, already during the design workshops. The digital representation of the haptic design of the modeling method is transformed into a language model that can be (a) verified against various platforms, (b) elevated semantically to extend concepts defined following open linked data and (c) generate library fragments as input to be combined in the modeling method.

Figure 2 depicts the proposed instantiation of the agile approach graphically. The design iteration is triggered by observations made in the system-under-study (for retrospective activities) or innovative ideas by the expert team of stakeholders. These experts are considered with different aspects in the modeling challenge and compose a multidisciplinary team.

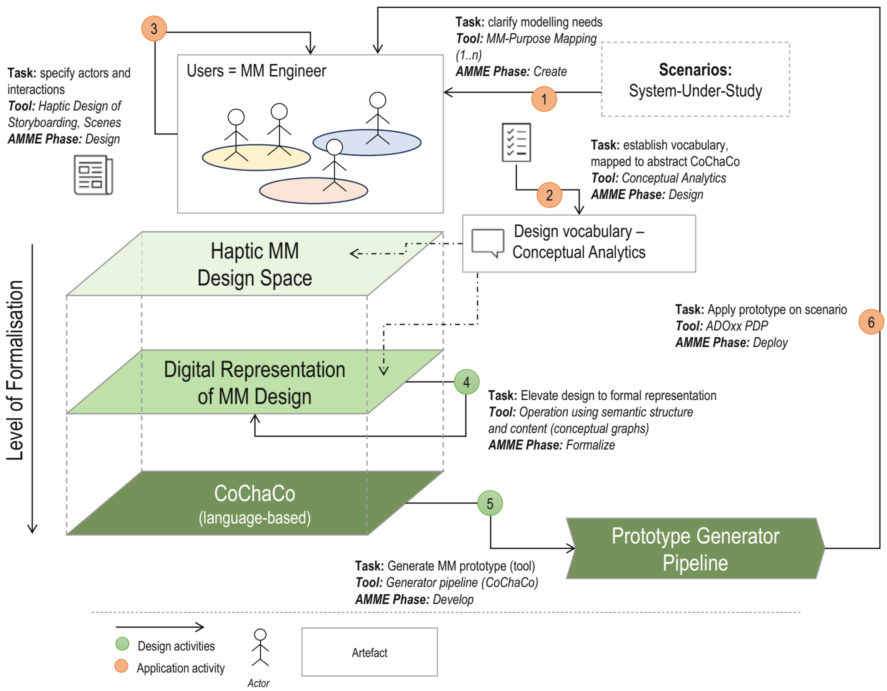


Fig. 2 Proposal: digital design thinking support in modeling method engineering

In a first step (1), the modeling purpose is jointly identified. Using as an analogy the Osterwalder's Business Model canvas in [21], and as a tool the "Modeling Method Purpose" mapping is proposed. Thus, the resulting canvas tool allows the stakeholders to define a) the purpose of the conceptual models to be created, the required inputs (service functionalities, data sources), the required conceptual elements and their relation. The definition is done in a high level of abstraction (textual description using sticky notes). This description and structuring outcome provide input to the second step, namely, to derive the required design vocabulary. Text analytics mechanisms suggest concepts and domain context and trigger the graphical research for iconic representations. The vocabulary elements are printed and convey the semantics as the team has agreed upon. The recognition services are dynamically aligned (see [22] for implementation alternatives) to be used within the Scene2Model tool. In addition, the vocabulary configures the language generator. Each artefact is mapped to the specific modeling constructs of CoChaCo[3] as a specialization relation and therefore allows preliminary configuration of the transformation and generator pipeline.

Scene2Model supports the expert team in the third step: actors and interactions are discussed using haptic elements. As soon as an agreement is reached, a digital representation as a model is persisted using Scene2Model.

The team jointly develops interaction scenarios as scenes (and decomposition of scenes) defining the way how actors interact with the modeling constructs. Each storyboard represents as an outcome a solution model for the modeling purpose identified.

As the mapping to CoChaCo is defined, a language-based representation becomes feasible. Utilizing a representation as conceptual graphs (see prototype section in [16]), design operations on structure and content are possible. These operations support the semantic elevation, service/functionality discovery and dynamic binding and consequently verification of the design on a formal level.

The last two phases are concerned with development and deployment. The open-source ADOxx metamodeling platform has proven to be flexible to support adaptive metamodel implementation. This means that the formal representation of the metamodel, the functional service stubs and interaction logic are provided to a generator pipeline (5) implemented as a continuous integration and deployment pipeline.[4] The outcome is a prototypical tool that can be installed and evaluated by the expert team (6).

[3] Accessible online at https://www.omilab.org/activities/cochaco/

[4] Further details available online at https://www.adoxx.org/documentation/60_packaging/2_packging_procedure.html. For the prototype, the AdoScript extension in Microsoft Visual Studio Code and the code repository runners at https://code.omilab.org where used.

4 Proof-of-Concept Prototype

As a means to evaluate the idea to utilize design thinking for modeling method engineering, a proof-of-concept prototype has been realized. The prototype focuses on an exemplifying scenario of a novel modeling method for geolocation modeling in a smart city environment. The objective of the proof-of-concept is to demonstrate its applicability.

Proof-of-Concept Setting. For the proof-of-concept the following setting is assumed: expert stakeholders (city tourism board, public transportation planning and external mobility providers and citizens) aim to optimize the public transport infrastructure, specifically for accessibility of touristic points-of-interest using multi-modal mobility services. The experts have recognized the value of modeling to evaluate in a forward-looking manner the feasibility of planning decisions taken by the city council.

1. **Clarify Modeling Needs.** The experts jointly assess the modeling needs with the following outcome. Tool support is provided to them during the brainstorming exercise.

   Figure 3 exemplifies the modeling method canvas: modeling needs are identified in the middle column, pre-existing data structures and processing techniques as candidates on the left and resulting constructs are listed on the right.
2. **Establish Design Vocabulary.** Based on this ideation and sorting results the concepts of the design vocabulary can be derived, e.g. the *city planning expert* will utilize a set of *stations* and *POIs* to determine the *effectiveness* of the *public transportation system* based on their *location.* Modeling constructs are marked italic above. Mapping those to a concrete notation/recognition service (for the design workshop) and CoChaCo constructs (as input for the generator) results in the visualization of the vocabulary as shown in Fig. 3.
3. **Design Storyboards and Scenes (Actors and Interactions).** The expert participants iteratively discuss the interactions specific actors require from the modeling environment. The workshops are conducted using the vocabulary elements as haptic elements (printed/drawn). This supports on one hand the storytelling approach envisioned as well as common space for argumentation and elicitation. In case of missing haptic elements, the system allows for a dynamic adaptation. This means that additional constructs can be added and utilized dynamically.

   Finally, the experts come to a common understanding and trigger the digitalization of the storyboard using the Scene2Model toolkit (see [23]). The digital design result using Scene2Model is visualized in Fig. 4 for the case "Location assessment". In this example, the actor requires a model that allows for a graphical mapping of *stations* and *POIs*, whereas the actual location is important and should be represented on a geographical map. Assessment techniques are required to evaluate the proximity of stations to POIs and vice-versa.

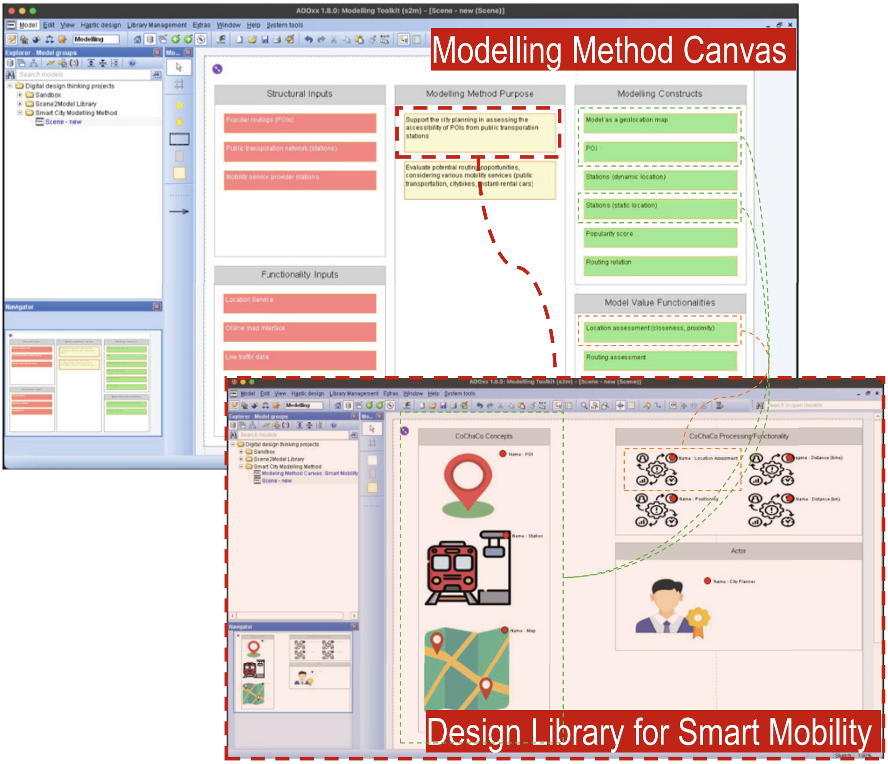


Fig. 3 Modeling Method Canvas (instantiated with Smart Mobility) and Mapping of Construct to CoChaCa resulting in the design library for the modeling method

4. **Formalisation and Functionality Alignment.** This step is concerned with the transformation of the digital design artefact to a formalized representation. Building upon the mapping towards CoChaCo and the results in [16, 18, 24] the formalization is performed. Formalization is required to (a) elevate characteristics based on concrete service functionalities (input/output relations) discovered, (b) the generator to translate the platform-independent into a platform-specific representation and (c) verify the syntactic completeness of the platform-specific model. Manual intervention by a modeling method engineer is currently required as the semantic distance only allows for a formalization as a skeleton structure. Using conceptual structures (formalized as conceptual graphs) are considered as a candidate to support design operations on semantic structure and content.
5. **Modeling Method Prototype Generator.** Using the transformation service developed in [16], the CoChaCo representation is transformed in a formal language representation, used to generate an ADOxx Library in ADOxx Library Language (ALL) format. The library serves as input for the continuous integration and deployment pipeline and dynamically deploys the prototype as an installation package. The prototypical realization utilizes the adaptive metamodel capabilities of the platform.

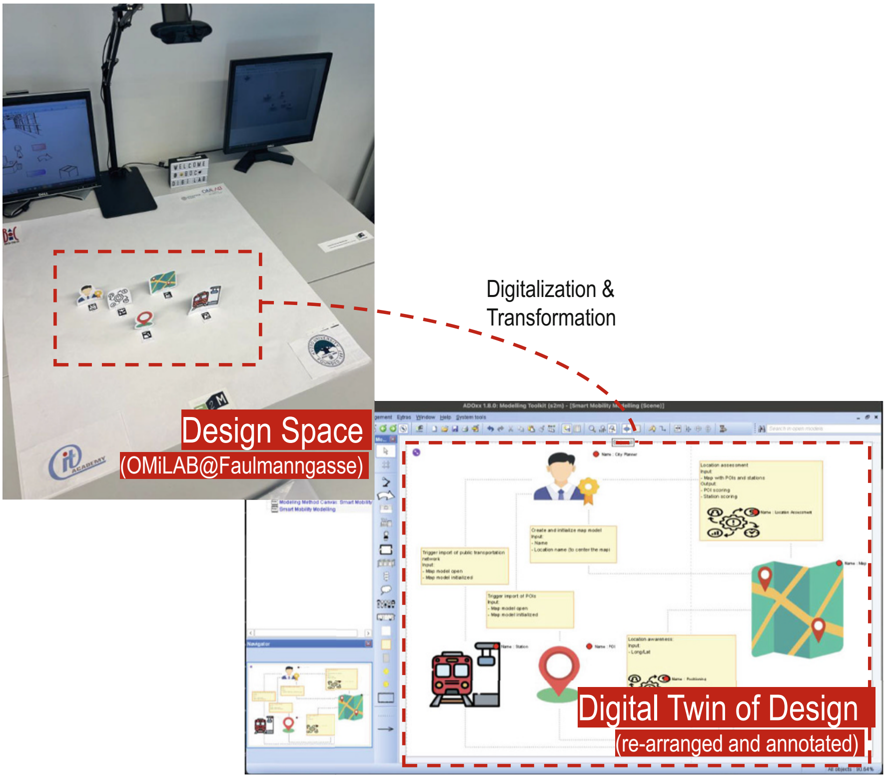


Fig. 4 Example Design and Digital Twin: Model-based Location Assessment

The expert stakeholder team has now the possibility to test and evaluate their design decision in the prototype tool and trigger another design iteration as outlined in Fig. 2, in case required.

5 Conclusion

In this paper a new design technique for modeling methods has been introduced, as an instantiation of the Agile Modeling Method Engineering practice defined by Prof. Karagiannis in [2].

The technique has been specified based on observations in the various OMiLAB community projects. It could be observed that tooling/operationalization of modeling methods has become an important aspect, but the knowledge of how to create modeling tools is challenging, specifically in cases where an early prototype is required to evaluate novel concepts and interaction flows. Using the design technique, it is assumed that in the early phases of the specification of a modeling method, tangible results in the form of haptic designs and corresponding prototypes, enable an increased capability to assess whether ideas are feasible and should be further refined.

Additionally, it can be observed that the work on conceptual modeling has become a cross-cutting issue that requires input from experts that have been unrelated to the world of models, metamodels and modeling methods in the past. Means are therefore needed to involve and externalize expertise and propose constructs that are value-adding. This requires novel interface and interaction patterns as proposed in this paper and integration with e.g. low-code/no-code approaches.

The current version of the approach is still in an early phase. Further work is required, specifically to investigate design scenarios that can be envisioned (retrospective design and alignment, composition, and re-use of existing modeling methods), external tool/service support for semantic elevation and functionality extensions.

References

1. Buchmann, R.A.: The purpose-specificity framework for domain-specific conceptual modeling. Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools, pp. 67–92 (2022). doi:10.1007/978-3-030-93547-4_4/
2. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics - PCI '15, New York, NY (2015). doi:10.1145/2801948.2802040
3. Karagiannis, D., Burzynski, P., Utz, W., Buchmann, R.A.: A metamodeling approach to support the engineering of modeling method requirements. In: Proceedings of the IEEE International Conference on Requirements Engineering, pp. 199–210 (2019). doi:10.1109/RE.2019.00030
4. ADOxx Metamodelling Platform, https://www.adoxx.org/, last accessed 2024/01/20
5. Karagiannis, D., Buchmann, R.A., Boucher, X., Cavalieri, S., Florea, A., Kiritsis, D., Lee, M.: OMiLAB: a smart innovation environment for digital engineers. In: IFIP Advances in Information and Communication Technology, pp. 273–282. Springer Science and Business Media Deutschland GmbH (2020). https://doi.org/10.1007/978-3-030-62412-5_23
6. Object Management Group (OMG): BPMN 2.0, http://www.omg.org/spec/BPMN/2.0/, last accessed 2024/01/17
7. Object Management Group (OMG): UML 2.5, http://www.omg.org/spec/UML/2.5/, last accessed 2024/01/17
8. Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.): Domain-Specific Conceptual Modeling. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-39417-6
9. Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W.: Domain-specific conceptual modeling: concepts, methods and ADOxx tools. Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools, pp. 1–656 (2022). doi:10.1007/978-3-030-93547-4/COVER
10. Strahringer, S.: Zum Begriff des Metamodells. Techn. Hochsch., Inst. für Betriebswirtschaftslehre
11. Wand, Y.: Ontology as a foundation for meta-modelling and method engineering. Inf. Softw. Technol. **38**, 281–287 (1996). https://doi.org/10.1016/0950-5849(95)01052-1
12. Höfferer, P.: Achieving business process model interoperability using metamodels and ontologies. In: Proceedings of the 15th European Conference on Information Systems, ECIS 2007, pp. 1620–1631 (2007)
13. Jeusfeld, M.A., Jarke, M., Mylopoulos, J.: Metamodeling for method engineering Choice Rev Online 47, 47-3853-47-3853 (2010). doi:https://doi.org/10.5860/choice.47-3853

14. Mylopoulos, J.: Conceptual modelling and Telos. Conceptual Modeling, Databases, and Case An integrated view of information systems development, pp. 49–68 (1992)
15. Utz, W.: Design metamodels for domain-specific modelling methods using conceptual structures. In: CEUR Workshop Proceedings (2018)
16. Utz, W.: Design of Metamodels for Domain-Specific Modelling Methods using Conceptual Structures (2020). https://doi.org/10.25365/thesis.63113
17. Fill, H.-G., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. Enterprise Modelling and Information Systems Architectures, 8 (2013). doi:https://doi.org/10.1007/BF03345926
18. Döller, V., Karagiannis, D.: Formalizing conceptual modeling methods with MetaMorph. Lecture Notes in Business Information Processing. **421**, 245–261 (2021). https://doi.org/10.1007/978-3-030-79186-5_16
19. Brown, T.: Design thinking. Harv. Bus. Rev. **86**, 84 (2008)
20. Brinkkemper, S., Saeki, M., Harmsen, F.: Meta-modelling based assembly techniques for situational method engineering. Inf. Syst. **24**, 209–228 (1999). https://doi.org/10.1016/S0306-4379(99)00016-2
21. Osterwalder, A., Pigneur, Y.: Business Model Generation - Canvas
22. Muck, C.: A Recognition Service for Haptic Modelling in Scene2Model. (2023)
23. Miron, E.-T., Muck, C., Karagiannis, D.: Transforming Haptic Storyboards into Diagrammatic Models: The Scene2Model Tool. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019). https://doi.org/10.24251/hicss.2019.067
24. Visic, N., Fill, H.G., Buchmann, R.A., Karagiannis, D.: A domain-specific language for modeling method definition: From requirements to grammar. In: Proceedings - International Conference on Research Challenges in Information Science (2015). doi:https://doi.org/10.1109/RCIS.2015.7128889

Process Oriented Knowledge Engineering: Reflections and Project Experiences

Robert Woitsch

Abstract Process oriented knowledge management started as a research topic around 2000. 2023 was counted year 1 of ChatGPT, this paper reflects how process oriented knowledge management evolved over the time by reflecting a series of research projects—mainly in the context of the EU Research Frame Programmes FP6, FP7, H2020 and HEU. A research topology consisting of (a) application scenario level, (b) model-based knowledge engineering level and (c) enabling IT environment level is used to structure the different initiatives. Three EU projects are introduced in more detail, discussing (1) Process Optimization in the FAIRWork project (HEU), (2) Process Digitization in the Change2Twin project (H2020) and (3) Process Deployment in the CloudSocket project (H2020). The corresponding support of knowledge management and engineering as well as the enabling IT-infrastructure are introduced. The reflection reasons that although the technology and algorithms massively evolved over the years, the underlying meta model to link knowledge management and engineering to organizational structures such as processes to ensure a targeted support has proven to be an appropriate assumption over the years.

Keywords Business process · Knowledge engineering · Microservices · Process optimization · Process digitization · Process deployment

1 Introduction

In the early 2000 the "Business Process Oriented Knowledge Management" approach was introduced to align knowledge management or knowledge engineering with organizational goals [1–3]. 2023 is year "1" of ChatGPT which promises to be among the game changing technologies in the context of knowledge

R. Woitsch (✉)
BOC Products & Services AG, Vienna, Austria
e-mail: robert.woitsch@boc-group.com

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_13

management. Hence, this paper reflects the evolution of process oriented knowledge management and engineering by discussing selected research projects—mainly in the context of the EU-Research Frame Programmes from FP5 to HEU.

Business Process Management (BPM) was introduced in the 1980ties and since then evolved to become commodity in today's business. With BPMN, a common standard notation for business process management has been established.

Knowledge Management and Engineering was distinguished between the more human-oriented approach—which was considered as "Knowledge Management" and the more computer-oriented approach—which was considered as "Knowledge Engineering" [4]. Those two approaches continuously merged, especially as with the evolution of the former "Computer Supported Cooperative Work" (CSCW) towards Web 2.0 into today's social media the two approaches can hardly be isolated.

Technological Infrastructure has been changed significantly. Service-based approaches are now commodity like Software-as-a-Service (SaaS) or Microservices. Low-code /no-code platforms support the integration of (a) data and its corresponding semantic, (b) the algorithms as well as (c) the user interaction.

In the following the research topology is introduced to identify the layers (a) application scenario, (b) conceptual modelling and (c) enabling IT environment in Chap. 2. In Chap. 3, we describe three research projects providing knowledge engineering for processes: First, to support process optimization, second to support process digitization, and third to support process deployment. Chapter 4 presents the corresponding model-based Knowledge Engineering (KE) approaches. Chapter 5 introduces the implementation of corresponding software environments. Finally, Chap. 6 concludes with a reflection.

2 Research Topology

We depicted three different horizontal layers and two vertical pillars, where the left pillar represents the open world and the right pillar the reflection of our research interest by applying different methodologies on each level (Fig. 1).

The **"Application Scenario Layer"** defines the domain. The applied methodology requires to understand the use case needs, abstract the generic challenge, and map it to capabilities that are provided by available modelling environments, like ADONIS [5]. This is an intellectual and interactive process performed by domain experts. It may range from selecting: (a) one capability, e.g. the simulation of a process, (b) several capabilities, e.g. the process deployment in combination with monitoring, or (c) a set of alternatives are worked out to implement new capabilities.

The **"Conceptual Modelling Layer"** designs model-based knowledge engineering solutions. The applied method is "Agile Modelling Method Engineering (AMME)" [6]. As sources we use (a) ADOxx.org [7], a community of more than 5.000 developers that provides open source features, (b) the OMiLAB [8] network with about 50 model-based solutions published in [9, 10], as well as (c)

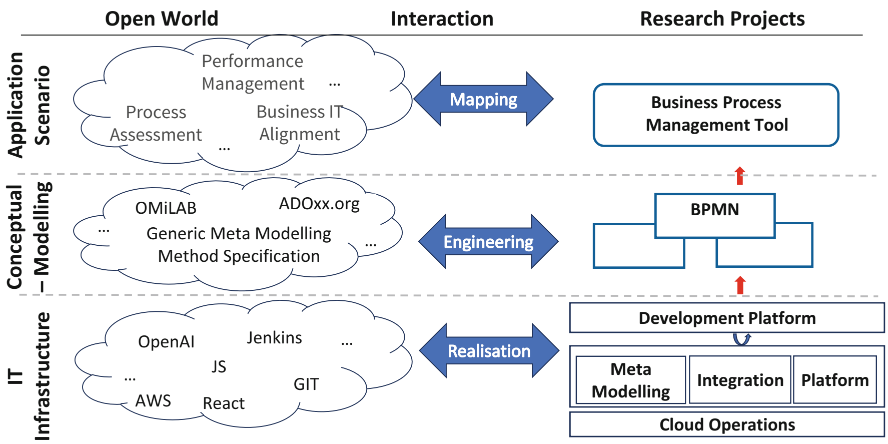


Fig. 1 Introduces the three layers (**a**) application scenario, (**b**) conceptual modelling and (**c**) enabling IT environment and the corresponding methodology that is applied to reflect and filter research initiatives for each layer

the principles of the Generic Meta Modelling Method Specification Framework (GMMSF) [11]. We apply rapid prototyping using the open accessible meta modelling platform ADOxx.

The **"IT Infrastructure Layer"** realizes functional capabilities of the meta modelling platform either by (a) selecting existing features, (b) configuring meta-concepts that enable the flexible re-usage and reconfiguration of functional capabilities, or (c) introducing a new functional capabilities by implementing them.

2.1 Applications Scenarios for "Process Oriented Knowledge Engineering"

There is a plethora of process-oriented scenarios like but not limited to continuous improvement, business IT alignment, business continuity management, process assessment, quality management, data protection, performance management, audit management, requirements management, process mining, ERP integration, or customer journey mapping [5]. Different scenarios in different domains have been addressed by research projects. In the public administration, processes were used to specify the public services of chamber of commerce's (LD-Cast [12]) or document immigration procedures (Immigration Policy 2.0 [13]). For training purposes of administration personnel, processes have been used as training platforms (LearnPAd [14]). In the construction domain the processes were used to simulate the usage of a building during design phase (Adapt4EE [15]) and during the renovation of a buildings processes were used to simulate expected progress and costs (BIMERR

[16]). To support the construction of railways, processes have been used to create a digital twin (COGITO [17]) to enable the execution of optimization algorithms.

Three selected projects are elaborated in more detail:

- **Process Optimization:** In the context of the project FAIRWork [18] the production processes are optimized by introducing knowledge-based dashboards.
- **Process Digitization:** In the context of the project Change2Twin [19] a digital twin was created by digitizing the paint production process.
- **Process Deployment:** In the context of the project CloudSocket [20], the business processes are modelled not only to be executable in a standard environment but also deployable in a special configured environment via a so-called "bundle" definition.

Based on these research projects, we observed the following characteristics:

- *simple design* to create common understanding and derive domain-specific semantic like in "Immigration Policy 2.0" and "LD-Cast",
- *acting as technological and content-wise integration* and collaboration platform like in "LearnPAd" and in "FOCUS" [21],
- using *processes for decision support* using process simulation, optimization and monitoring like in "DISRUPT" [22], or in "GO0DMAN" [23],
- *execution of processes* via workflows like in "CaxMan" [24] or "CloudSocket".

2.2 Conceptual Modelling for "Process Oriented Knowledge Engineering"

Graphical concept models can serve both purposes, the human and the computer interpretation. The graphical representation is more appropriate for intuitive human interpretation, whereas the conceptual and semantic representation is more appropriate for machine interpretation. Hence, concept models can act as a moderator between human-oriented and machine-oriented approaches as elaborated in the project plugIT [25]. Based on the work on meta model mapping [26], and meta model weaving (LearnPAd) we differentiate three scenarios for modelling KE approaches:

First, **Models are used to identify** the needs of KE. A semi-formal model like a process in Business Process Model and Notation (BPMN) standard is sufficient to identify, where and how KE has to contribute. We see this as a "**delegation**" towards the algorithm, as no further specification is performed.

Second, **Models are used to specify** the expected results of the KE algorithm. Models are used to describe the algorithm on an abstract level that enables domain-experts to describe the behavior without special KE skills. We use e.g. (a) BPMN to specify an abstract workflow (CloudSocket [20]), (b) decision tables to specify the behavior of a rule engine (FIT [27]), (c) semantic networks to represent relevant parts of an ontology (Change2Twin), (d) goal models to specify the behavior of multi-agents (eHealthMonitor [28]), (e) Case Management Model and Notation (CMMN) to represent appropriate training data to define the behavior of neural networks is currently elaborated in the FAIRWork project.

Third, **Models are used to configure** the KE algorithm. This approach we apply for symbolic AI approaches like rules, workflows, fuzzy-rules, semantic inference.

Aforementioned model-based approaches require a corresponding IT environment that enables the implementation of IT-based solutions.

2.3 IT-Infrastructure for "Process Oriented Knowledge Engineering"

We rely on the ADOxx platform [7], which provides a rich set on model-specific functional capabilities as well as the possibility to configure the platform using the meta model concept. In the following the most helpful characteristics in implementing rapid prototypes for research projects are introduced:

- **Inheritance of modelling objects** defines the structure, the semantic and the corresponding functionality of sub-classes. Inheriting a class from the "root" or inheriting the class from a "container class" will have different results. The class from the root will inherit the basic capabilities of a graphical model element, whereas the class inherited from the container class in addition also inherits the semantic of being a container that graphically groups modelling objects and has the relation called "is-inside". The inheritance speeds-up the provision of functionalities.
- **Coding and configuration support** eases the configuration of the platform. We observe that by providing configuration possibilities like e.g. AdoScripts [29], user interface dialogs, previews or test environments, the flexibility raises as non-software engineers can quickly produce rapid prototypes.

3 Processes from Selected Application Scenarios

We introduce three selected samples from EU projects that demonstrate process oriented approaches for (a) optimization, (b) digitization, and (c) deployment.

3.1 Process Optimization: FAIRWork Experience

FAIRWork is a HEU project which combines data and AI technology to optimize production processes. First, the production of electronic parts at Flextronics [30] and second the production of metal parts for cars at Stellantis [31].

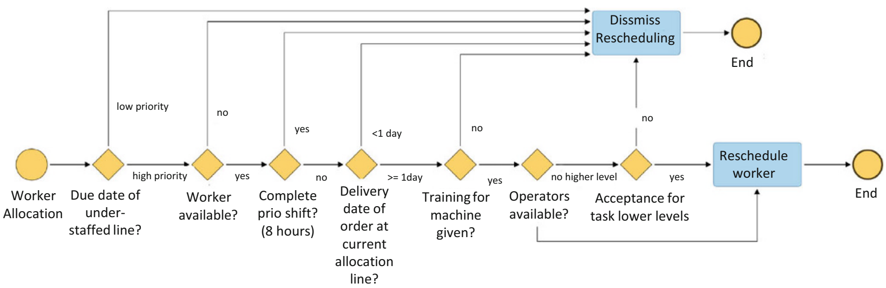


Fig. 2 Flextronic decision model of Worker Allocation in FAIRWork D5.1 [32] (page 70) indicates the needed steps to allocate the "most appropriate" worker to the production line

The approach is to optimize the overall process by introducing knowledge-based dashboards that apply KE algorithms to propose the "optimal" solution to the decision-maker. The introduced sample process addresses the pre-production in particular the worker allocation that has to consider parameters like worker capabilities for the line, the efficiency per worker per production line, deadlines and corresponding overtime issues, and worker preferences. The process collects the "screen play", which is the capability of a worker to work on certain production lines as well as the production plan or the worker preferences (Fig. 2).

The solution is a decision-support dashboard that shows the different parameters in different charts, where different KE algorithms can be used—also in combination—to continuously calculate the proposed optimal allocation (Fig. 3).

3.2 Process Digitization: Change2Twin Experience

Change2Twin is a HEU project performing digital twinning of production processes. In the following we introduce a sample of a Spanish paint production manufacturer called Graphenstone. The challenge is that due to increasing online orders, the acceptance of new orders depending on the status of the warehouse requires simulation of the production process using a real-time inventory of the warehouse. The digital twin is capable to simulate the orders considering the inventory of the warehouse.

- First, a **digital twin of the production process** was created. We used time stamps at relevant stages of the process to trace the start, the collection of material, the mixing, the laboratory tests and the filling.
- Second, the **real time inventory** of the warehouse was realized by introducing pure material slots, where the collection of the material was time-stamped with an RFID tag representing the process id.
- Third, the **product**—the painting bucket—**became digitized** by introducing a write-able RFID tag on the painting bucket label, where the production process id was stored. This enables to access the process data and laboratory test results.

Fig. 3 Mock-up of customizable User Interface FAIRWork D4.1 [33] (page 35) using different user interface templates, that present the different results from different algorithms

The process was modeled covering three production lines, the interaction with the warehouse, and the laboratory tests. The key stages of the process have been identified to digitize the status of the process by using time stamps (Fig. 4).

A physical experiment [34] has been developed for the chemical production process by abstracting the relevant stages of the paint production process to the generic and well-known sample of a "tea coocking process" (Fig. 5).

The ingredients are selected distinguishing basis material like hot water, product-specific materials like different flavors of tea, and product-independent supplements like sugar. Those ingredients are mixed, tested, and filled in small cups where each cup has a writeable RFID tag. Selected challenges of the production process e.g. how many teas can I produce in the next 2 h, how to ensure to have enough sugar, or how to calculate the waste of milk, can be easily reported, and different approaches were able to be demonstrated with varying capabilities. Finally, the factory implemented the setup of this experiment with selected digitizing technologies.

3.3 Process Deployment: CloudSocket Experience

CloudSocket was a H2020 project introducing the Business Process-as-a-Service (BPaaS), where a business process is defined not only on business level, but also on executable workflow level, and on deployment level. A BPaaS—e.g. sending

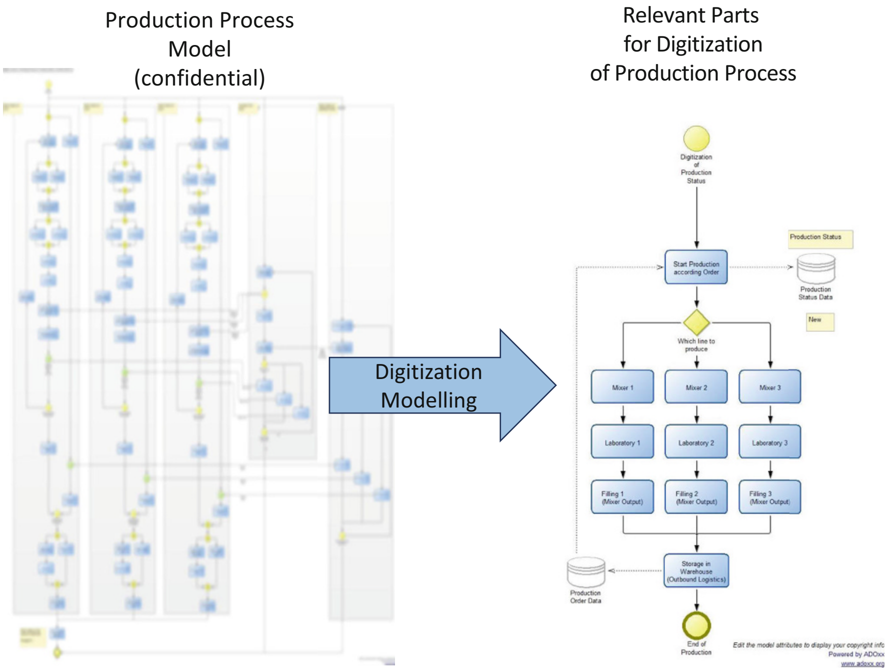


Fig. 4 The confidential production process—which is blurred for confidentiality reasons—has been modelled and relevant actions that are needed for digitization have been selected to perform time stamps [34]

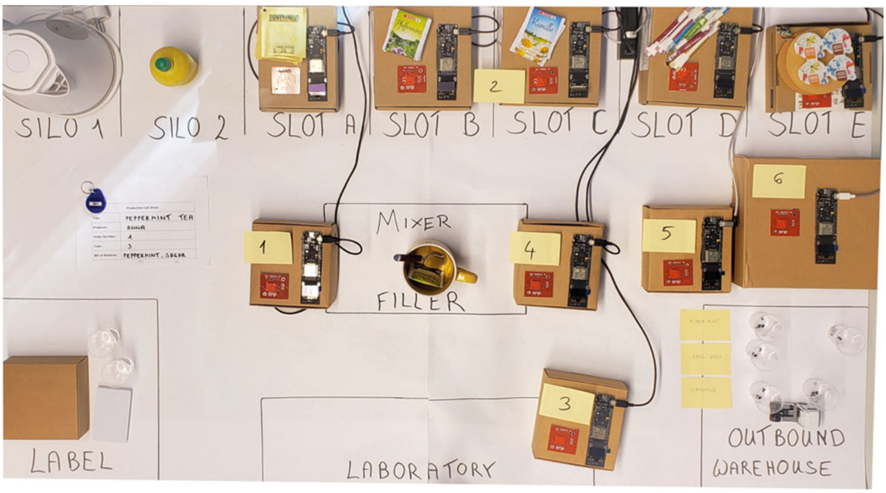


Fig. 5 Shows the physical experiment of the production process, whereases silo 1 and 2 represent the material that is not digitized, slot A to E are digitized with RFID readers, the production process is a RFID tag that is physically sticked on the piece of paper which is the production process description, and the different stages of the process are indicated on the flip chart

an invoice to a customer—can be implemented via various executable workflows, whereas one executable workflow can be offered in form of various deployed services with different Service Level Agreements (SLAs).

A BPaaS bundle contained all information to present a business process as a service on a marketplace. The user can view both—the domain specific process and the executable workflow. Deployment information was provided via Cloud Application Modelling and Execution Language (CAMEL) [35]. The marketplace displayed the price and corresponding SLAs including execution times as well as the location of the data processing. The result was that the business process "Sending and Invoice" was offered in the form of different services, once as cheap but unreliable bundle with data processing outside of Europe, and once as a costly, reliable bundle with guaranteed data processing at a certain location (Fig. 6).

Buying the BPaaS from the marketplace triggers the deployment of the bundle. The use of generic deployment formats enables the deployment in a multi-cloud environment. More details are provided in the publication collection on Zenodo [37]. Relevant prototypes can be found at the innovation shop provided by BOC [38] (Fig. 7).

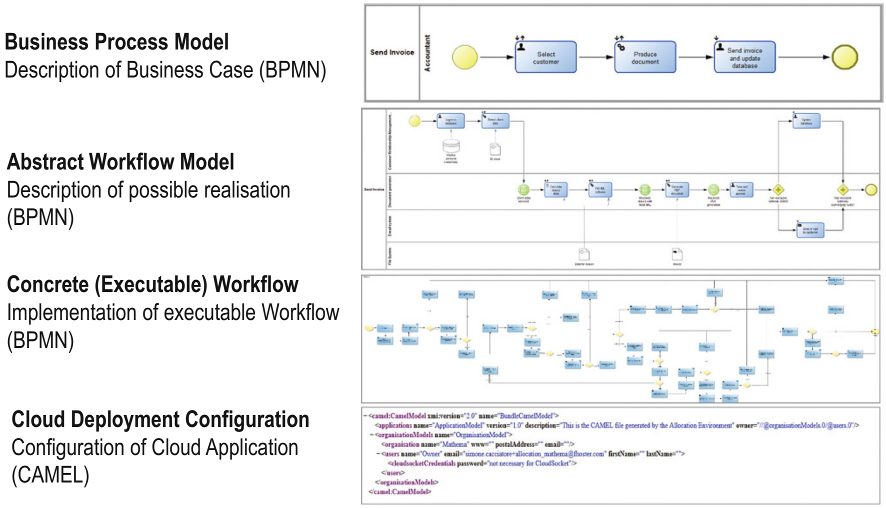


Fig. 6 Introduces the different levels of the BPaaS "Sending an Invoice" [36]. The business process model describes a simple invoice sending process in BPMN format. The abstract workflow describes the sequence of the workflow in a way that is understandable for domain experts. The executable workflow adds technical details. Finally, the cloud deployment script, that completes the deployment, adds information of the deployment of this BPaaS

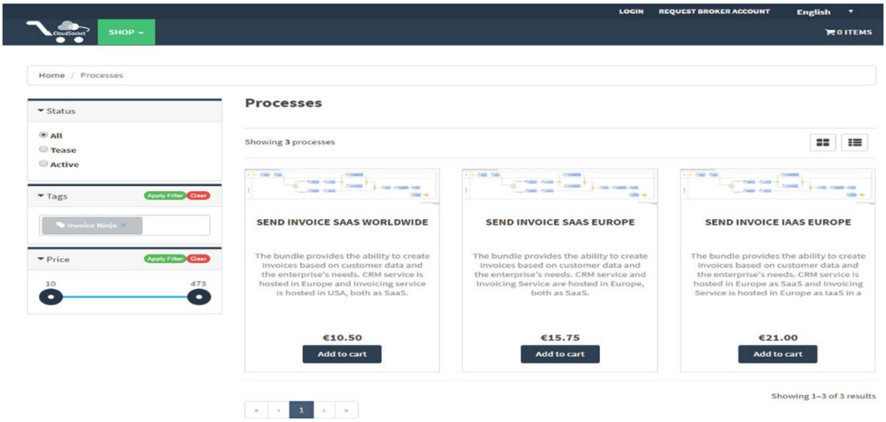


Fig. 7 Shows a Marketplace that offers different BPaaS bundle with same domain business process of sending an invoice but in different deployment settings

4 Model-Based Approaches for Knowledge Engineering

Each application scenario is supported by KE. We applied model-based KE to (a) identify the needs, (b) specify the expected results, or (c) configure the algorithms.

4.1 KE for Process Optimization: FAIRWork Experience

We propose a continuous knowledge-based decision making to achieve an overall process optimum—in our case, the allocations of workers to production lines. First, the data from the production plan and other sources are accessed. Then the decision logic is performed by an algorithm. Finally the results are presented in a dashboard. We tested various algorithms in order to compare the different outcomes D4.2 [39].

- **Rule-Based mechanisms using DMN** introduced a decision tree to allocate the worker. Different allocations strategies exist, e.g. starting with the less experienced workers, or starting with the preferences of the workers. Here, DMN was used to configure the rule-engine with executable DMN files.
- **Fuzzy-Rules extending DMN models** made the decision tree more flexible thanks to different dependency level. The results can be transparently traced but the final configuration of the approach has to be performed in the corresponding fuzzy logic library. Here we used a specification approach.
- **Multi Agent based approach** uses negotiation between agents for each worker, the lines and the production plan. This approach is currently modelled on identification level using the decision tree as an input for manual programming the negotiation behavior of the agents.

- **Reinforcement Learning** uses sample data for training. The challenge is therefore to identify the correct training data. Currently, we stay on identification level but assume that modelled training and test data may improve the algorithm.

4.2 KE for Process Digitization: Change2Twin Experience

The challenge is to ensure the correctness of the real-time inventory of the warehouse in combination with the production process simulation considering possible production failure or inconsistencies in the warehouse inventory. We propose patterns to digitize the inventory of the warehouse ranging from expensive and exact to cheap and imprecise approaches. In collaboration with the project CALIBRaITE [40], the trustworthiness of a dashboard representation was introduced. Depending on the following different patterns, the dashboard indicated not only the data but in addition the corresponding trust level. The following patterns have been applied [41]:

- **No digitization pattern**: For material which is ordered regularly we proposed to not digitize the inventory. There are some basic liquids that are needed for the production, which are regularly filled independently of the concrete production plan.
- **Calculation pattern**: This pattern counts how often a material has been picked up using the time stamps of the production process id. This requires discipline in using the material and estimating the lost material. We use process simulation considering probability deviation for the material loss. The trust level is medium.
- **Measuring pattern**: A precise approach is to measure the weight of pure material slots and track the reduction of weight in combination with the process ids. This approach is expensive as each slot need individual scales as additional sensors. This pattern achieves the maximum of the trust-level.
- **Approximation patterns**: Approximations are performed using (a) image recognition to approximate the status of the material and (b) distance sensors measuring the height of the material stacks and estimate the amount. The result of such approximations heavily depends on the selected algorithm and the training data. This pattern was not selected during the project, as the decision maker needs to gain trust. The trust-level for this pattern varies.

The physical experiment of this process allowed to test different KE algorithms for simulation, for the assessment of the material as well as for plausibility checks.

4.3 KE for Process Deployment: CloudSocket Experience

To support the modelling and execution of a BPaaS bundle, several algorithms have been provided [38]:

- **Context-adaptive Questionnaire for Business-IT Alignment** [42]: Each BPaaS needs allocation of concrete Software-as-a-Service (SaaS) offerings which perform only a part or the whole process. This allocation was supported by semantic lifting of both, the process model and the SaaS offerings. We proposed a questionnaire for the annotation. The user filled out a questionnaire, while annotating the model. The questionnaire used an entropy-based algorithm to find a complete service annotation with a minimum of questions.
- **DMN to CAMEL Mapping [**43**]:** Deployment scripts used decision models in form of DMN rules introduced more flexibility into the deployment strategy. A simple decision layer was introduced on top of the cloud application description within the deployment model, which abstracts from the actual deployment language and allows to assemble the deployment model from existing fragments.
- **Smart Business Intelligence Analysis Tool [**44**]:** To enable cross-layer monitoring we used semantic mappings between the different layers. The abstraction of business-related indicators such as costs, performance or location of deployment, can not only be collected on the cloud infrastructure but must be calculated across several layers up to the business process model layer. We proposed to include semantic inference to bridge the semantic distance of the different layers.

Aforementioned samples of KE algorithms and their model-based approaches are selected prototypes that have been developed in the aforementioned EU research projects.

5 Implementation Approach

The IT environment provides domain-specific and model-based IT Solutions. We used the metamodelling platform ADOxx to develop research prototypes. Our research focus in the mentioned project was to extent or complement the environment around ADOxx as well as explore new technologies. In the following we introduce some results:

- **Domain specific Services for Processes:**
  - **Process Verification** [45] was implemented in CloudSocket based on the LOLA approach, where processes are mapped to Petri-Nets and then analyzed according to their correctness e.g. deadlocks, endless-loops, or formal correctness. This ensured only executable workflows are specified by domain experts.
  - **Process Simulation** [46] was implemented in BIMERR based on a Petri-Net Simulation engine, where BPMN was mapped to Petri-Nets. We introduced a multi-parameter pre-processing for the simulation with individual probability functions. Different influence factors like e.g. weather, season, failure were individually described with probability functions and pre-processed before their usage during simulation. This ensured that the probabilities could be modelled in more detail and the simulated prediction was more appropriate.

- **Integration Framework to integrate Microservices** [47]: Current research elaborates the development of a low-code/no-code platform called OLIVE that provides (a) integration of data via a self-developed connector framework, (b) integration of services via the workflow engine Netflix Conductor, and (c) integration of user interfaces via the React UI framework.
- **Physical modelling extending digital modelling** [48]: We observed design thinking approaches, in particular the Scenes approach from SAP [49], and the corresponding work of Scene2Model™ from OMiLAB NPO and adapted the technology, which we call "Scenario Scanner" to apply them for process modelling.
  - The scenario scanner was used in workshops to work out **use case specific processes**. This enabled us to define processes in a workshop setting where most of the people are unaware of the process notation BPMN. The resulting sketches are published in FAIRWork D2.1 [50].
  - The scenario scanner was used in form of **table-top exercises** to test a complex resource allocation in the COGITO project. The construction site of a railway was first modelled, and then the resource allocation was calculated. The core team could perform a table-top experiment by physically walking though the construction process using the picture of the construction site that has been filmed from a drone. This hands-on exercise generated common understanding of the core team and raised awareness on potentially critical situations.
  - Third, we used an extended version including RFID tags of the scenario scanner to **demonstrate** a new painting production process at Graphenstone in the project Change2Twin, which was introduced in Sect. 3.2.

Aforementioned samples showed the corresponding extensions that has been provided on several levels on ADOxx. Either in form of domain-specific services, in form of an integration framework or in form of a physical extension for the digital environment. A generic approach on how to integrate knowledge engineering algorithms in form of Microservices is published in the FAIRWork D4.1 [33] and D4.2 [39].

6 Reflection

Reflecting current research projects using processes, model-based knowledge engineering and new IT-platforms, we can see a massive evolution on the technology and algorithms. AI algorithms coming from AWS, Azure, Alibaba, or Google are available for interested researchers. Software Engineers use AI-based tools like Tabnine, GitHub CoPilot, or AWS Code Whisperer to speed up coding. ChatGPT and alternatives like YouChat, Jaspe Chat, Microsoft Bing Chat, Google Bard, or Neuroflash are likely to quickly enter everyday's life of an organization.

We still believe that concepts and technologies should be separated. The intention of the algorithm including issues like trust, transparency, reliability or the capability of being revision-able need to be identified first, before the appropriate algorithm is selected. The concrete values of appropriate models have been studied in the project complAI [51], where models are used to ensure an intended behavior of AI algorithms. Currently, we observe challenges like (i) how to introduce AI into an organization, (ii) how to deal with ethical issues like privacy or biased results or (iii) how to ensure transparency and trust. The approach presented in this paper—to align KE algorithms to concrete organizational challenges using a model-based approach—seems still to be a reliable and practical approach. Furthermore, we see the challenge that algorithms will be provided for all layers introduced in this paper: (a) the application scenario layer, the (b) conceptual modelling layer and the (c) IT-Infrastructure layer. Based on the reflection of the past year, the evolution of AI is promising and has the potential to become a game-changer in several important domains.

References

1. Woitsch, R.: Process Oriented Knowledge Management: A Service-Based Approach. Dissertation, Institut für Knowledge and Business Engineering, Universität Wien, Wien (2004)
2. Karagiannis, D., Telesko, R.: The EU-project PROMOTE: A process-oriented approach for knowledge management. In Proceedings of the Third Int. Conference on Practical Asprects of Knowledge Management (PAKM), Basel (2000)
3. Abecker, A., Hinkelmann, K., Maus, H., Müller, H.J.: Geschäftsprozess-orientiertes Wissensmanagement. Springer, Berlin (2022)
4. Karagiannis, D., Woitsch, R.: Knowledge engineering in business process management. In vom Brocke, J., Rosemann, M. (Eds.), Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture, Springer, Berlin, Heidelberg, pp. 463–484 (2010). doi: 10.1007/978-3-642-01982-1_22
5. BOC, https://www.boc-group.com/de/adonis/, access: 31.01.2024
6. Karagiannis, D.: Conceptual modelling methods: the AMME agile engineering approach. In: Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W. (eds.) Domain-specific conceptual modeling. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-93547-4_1
7. ADOxx.org, The ADOxx metamodelling platform – reference webpage. http://www.adoxx.org/, access: 31.01.2024
8. The OMiLAB Community: Development of conceptual models and realization of modelling tools within the ADOxx meta-modelling environment: a living paper. In: Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W. (eds.) Domain-Specific Conceptual Modeling. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-93547-4_2
9. Karagiannis, D., Mayr, H., Mylopoulos, J. (eds.) Domain-Specific Conceptual Modeling, Springer, Cham (2016). ISBN: 978-3-319-39416-9
10. Karagiannis, D., Lee, M., Hinkelmann, K., Utz, W. (eds.) Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools. Springer International Publishing, Cham (2022). ISBN: 978-3-030-93546-7
11. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) E-Commerce and Web Technologies. EC-Web 2002 Lecture Notes in Computer Science, vol. 2455. Springer, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-45705-4_19

12. Missikoff, M., Taglino, F.: A semantic cooperation and interoperability platform for the European chambers of commerce. In: Vitvar, T., Peristeras, V., Tarabanis, K. (eds.) Semantic Technologies for E-Government. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-03507-4_6
13. Polemi, N., Pentafronimos, G., Ntouskas, T.: IMP2.0 Migration collaborative platform and e-services. In: Skersys, T., Butleris, R., Nemuraite, L., Suomi, R. (eds.) Building the e-World Ecosystem. I3E 2011. IFIP Advances in Information and Communication Technology, vol. 353. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-27260-8_4
14. De Angelis, G., Pierantonio, A., Polini, A., Re, B., Thönssen, B., Woitsch, R.: Modeling for learning in public administrations—the learn PAd approach. In: Karagiannis, D., Mayr, H., Mylopoulos, J. (eds.) Domain-Specific Conceptual Modeling. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39417-6_26
15. PONENCIA. "Energy Performance Using Open Studio and Business Process Data". SEMANCO Semantic tools for Carbon Reduction in Urban Planning. Workshop in Barcelona, Analysing and visualising energy related data in our buildings, towns and cities, La Salle Campus Barcelona, Spain, 11–12 April 2013. En colaboración con M. Eguaras y M. Vidaurre. https://hdl.handle.net/10171/36822
16. BIMERR: BIM-based tools for Energy-driven renovation of existing residences, https://bimerr.eu/
17. COGITO: materialising the digitalisation benefits for the construction industry to unleash the untapped potential in productivity improvement and increased safety. https://cogito-project.eu/
18. FAIRWork: Flexibilization of complex Ecosystems using Democratic AI based Decision Support and Recommendation Systems at Work. https://fairwork-project.eu/
19. Change2Twin: Bringing Digital Twins to Manufacturing SMEs. https://www.change2twin.eu/ , access: 31.01.2024
20. Woitsch, R., Utz, W. Business Process as a Service Model Based Business and IT Cloud Alignment as a Cloud Offering, Oct. 2015. doi: https://doi.org/10.5281/zenodo.35581
21. FOCUS: Foresight security scenarios: Mapping research to a comprehensive approach to exogenous EU roles. https://cordis.europa.eu/article/id/147119-improving-europes-security-responses
22. Eirinakis, P., et al.: A proposal of decentralised architecture for optimised operations in manufacturing ecosystem collaboration. In: Camarinha-Matos, L., Afsarmanesh, H., Fornasiero, R. (eds.) Collaboration in a Data-Rich World. PRO-VE 2017. IFIP Advances in Information and Communication Technology, vol. 506. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65151-4_12
23. Camarinha-Matos, L.M., Rocha, A.D., Graça, P.: Collaborative approaches in sustainable and resilient manufacturing. J. Intell. Manuf. (2022). https://doi.org/10.1007/s10845-022-02060-6
24. Efendioglu, N., Woitsch, R.: A modelling method for digital service design and intellectual property management towards industry 4.0: CAxMan case. In: Hara, Y., Karagiannis, D. (eds.) Serviceology for Services. ICServ 2017 Lecture Notes in Computer Science, vol. 10371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61240-9_15
25. Woitsch, R., Karagiannis, D., Plexousakis, D., et al.: Business and IT alignment: the IT-Socket. Elektrotech. Inftech. **126**, 308–321 (2009). https://doi.org/10.1007/s00502-009-0660-2
26. Kühn, H.: Methodenintegration im Business Engineering. Dissertation, Institut für Knowledge and Business Engineering, Universität Wien, Wien 2004
27. Karagiannis D., Utz W., Woitsch R. , Leutgeb A.: Business processes and rules: An eGovernment case-study, Published in AAAI Spring Symposium: AI 2008,
28. Kirn, S., Murray, J., Premm, M., Schüle, M., Widmer, T.: Towards a research framework for multiagent organizations. Proceedings of the International Symposium on Service Science (ISSS), Leipzig (2013)
29. AdoScript. https://marketplace.visualstudio.com/items?itemName=ADOxxorg.adoxx-adoscript, accessed: 31.01.2024
30. Flextronic. https://flex.com/, accessed: 31.01.2024
31. Centro Ricerche Fiat S.C.p.A. http://crf.it, accessed:31.01.2024

32. FAIRWork Consortium Deliverable D5.1 DAI-DSS FAIRWork Knowledge-Base at Use Case Site. https://fairwork-project.eu/deliverables/
33. FAIRWork Consortium Deliverable D4.1 DAI-DSS Architecture and initial Documentation and Test Report. https://fairwork-project.eu/deliverables/
34. Physical Experiment. https://innovation-laboratory.org/experiments/paint-production/overview/#, accessed: 31.01.2024
35. CAML. https://caml.inria.fr/, accessed: 31.01.2024
36. CloudSocket project, www.cloudsocket.eu, CloudSocket Webinar: Digitalize your Business (Olten, 15.12.2017). https://www.youtube.com/watch?v=Chqd7tVxb90, minute 7:10, accessed: 31.01.2024
37. CloudSocket. https://zenodo.org/search?q=cloudsocket, accessed: 31.01.2024
38. CloudSocket Innovation Shop. https://site.cloudsocket.eu/cloudsocket-innovation-shop, accessed: 31.01.2024
39. FAIRWork Consortium Deliverbale D4.2 Initial DAI-DSS Prototype. https://fairwork-project.eu/deliverables/, access: 31.01.2024
40. Peter, F., Mirnig A. G. , Damiano, F., Johann, S., Diamond, F. I., Manfred, T.: Effects of reliability indicators on usage, acceptance and preference of predictive process management decision support systems, In Quality and User Experience, Springer, Volume 7, article number 6 (2022)
41. Sumereder, A., Woitsch, R.: Digitization Principles for Application Scenarios towards Digital Twins of Organizations, in: eccomas2022. https://www.scipedia.com/public/Sumereder_Woitsch_2022a
42. Hinkelmann, K., Gerver, A., Karagiannis, D., Thoenssen, B., van der Merwe, A., Woitsch, R.: A new paradigm for the continuous alignment of business and IT: combing in enterprise architecture modelling and enterprise ontology. Comput Ind. **79**, 77–86 (2015). https://doi.org/10.1016/j.compind.2015.07.009
43. Griesinger, F., Seybold, D., Domaschka, J., Kritikos, K., Woitsch, R.: A DMN-based approach for dynamic deployment modelling of cloud applications. In: Conference Proceeding of European Conference on Service-Oriented and Cloud Computing (ESOCC), pp. 104–111. Springer (2016)
44. Kritikos, K., Plexousakis, D., & Woitsch R: Towards semantic KPI measurement. Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER), ACM, April 2017, pp. 91–102, doi:10.5220
45. ADOxx.org Development Space: https://www.adoxx.org/live/adoxxweb-verification-details, accessed: 31.01.2024
46. ADOxx.org Development Space: https://www.adoxx.org/live/adoxxweb-simulation-details, accessed: 31.01.2024
47. Falcioni, D., Woitsch, R.: OLIVE, a model-aware microservice framework. In: Serral, E., Stirna, J., Ralyté, J., Grabis, J. (eds.) The practice of enterprise modeling. PoEM 2021 Lecture Notes in Business Information Processing, vol. 432. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91279-6_7
48. Woitsch R.: Industrial Digital Environments in Action: The OMiLAB Innovation Corner, In 13th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling, PoEM Workshop2020
49. SAP. https://experience.sap.com/designservices/resource/scenes, access: 31.01.2024
50. FAIRWork Consortium Deliverable D2.1 Specification of FAIRWork and initial DAI-DSS Architecture: https://fairwork-project.eu/deliverables/, access: 31.01.2024
51. Woitsch, R., et al.: Collaborative model-based process assessment for trustworthy AI in robotic platforms. In: Gerber, A., Hinkelmann, K. (eds.) Society 5.0. Society 5.0 2021. Communications in Computer and Information Science, vol. 1477. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86761-4_14

Modular Metamodel Engineering: Why, What, How and What's Next?

Srdjan Zivkovic

Abstract Numerous standard and domain-specific modeling languages have been designed to facilitate the modeling of enterprise and software systems. Modeling language definition is a complex engineering task. Metamodeling is an established, practical yet rigorous formalism for language definition, which employs the metamodel as a pivotal engineering artifact. Modular metamodel engineering presents a systematic approach to metamodeling, incorporating concepts of modularization and composition to enhance reusability, flexibility, and efficiency in metamodel definition. It introduces the concept of reusable metamodel fragments, accompanied by a comprehensive set of metamodel composition operators for combining these fragments. This paper provides an overview of the modular metamodel engineering approach. Additionally, it delves into potential applications of microservices in the context of modeling method microservices architecture and discusses the potential application of generative AI, shedding light on its implications for advancing metamodeling practices.

Keywords Metamodeling · Metamodel composition · Metamodel modularization · Metamodel fragment · Modeling method microservice

1 Introduction

Modeling method engineering deals with the design, construction and adaptation of modeling methods and appropriate tools for model-based system analysis and development. It is a common sense that Domain-specific Modeling Languages (DSMLs) and corresponding methods satisfy the increasing pace of business requirements and specific problem domains better than the standard general-purpose methods. Demand for DSMLs, methods and corresponding tools is continuously

S. Zivkovic (✉)
BOC Information Technologies Consulting, Vienna, Austria
e-mail: srdjan.zivkovic@boc-group.com

H.-G. Fill, H. Kühn (eds.), *Metamodeling: Applications and Trajectories to the Future*, https://doi.org/10.1007/978-3-031-56862-6_14

increasing, as more stakeholders seek to leverage the power of model-based low-code/no-code approaches to abstract from code when describing and engineering complex systems. DSMLs in general focus on a narrow specific domain, however, complex enterprise systems often demand usage of DSMLs in combination forming hybrid modeling methods, to capture the system in a holistic way, but still retaining the specificness and expressivity for each modeling domain. Besides the hybrid characteristic, modeling methods undergo changes during their lifecycle, i.e. they evolve, to meet the continuous changes in business requirements, to incorporate gained domain knowledge over time, or to address the technological advancements of the underlying implementation system. For example, a company may adopt BPMN 2.0 [1] as a standard for business process modeling but may require company-specific extensions for process-based risk management. Such customization may involve introduction of additional risk-related properties to existing language entities, creation of new entities or even integration with proprietary languages to build a custom hybrid solution. Ideally, such custom extensions should be portable to the upcoming version of the base language, which for itself undergoes evolutive changes. The increasing demand for DSMLs, their hybrid nature, continuously evolving modeling requirements and underlying domain diversity pose new challenges to the engineering of modeling methods, in terms of efficiency, time-to-value, flexibility, domain and system complexity. Following the agility principles established in software engineering, Agile Modeling Method Engineering (AMME) [2] provides a framework to address these challenges in an agile way.

It has been recognized that *modular approaches* in software industry such as component-oriented development and micro-service architecture have been beneficial for designing complex systems, in terms of reusability, efficiency, enhanced flexibility, improved system understanding, standardization, maintainability, and scalability [3, 4]. A metamodel is a pivotal engineering construct in modeling language definition, that is used to define the language abstract syntax, for which one or more notations i.e. concrete syntaxes are defined [5]. Furthermore, in the context of modeling method definition, a metamodel serves as a data structure for method algorithms and mechanisms, and it represents the base vocabulary for modeling procedures [6]. To tackle the problem at its root, applying the principles of modular design and component orientation to metamodel definition (metamodeling) has been shown beneficial to address the challenges of modeling method engineering, in general. However, while metamodeling techniques provide powerful concepts for creating metamodels from scratch, they have been lacking concepts for more efficient metamodeling towards metamodeling-in-the-large [7], as well as for more flexible metamodel customization [8]. Modular metamodel engineering as defined in [9] has been an attempt to incorporate concepts of modularization and composition in metamodeling, with the purpose to enhance reusability and efficiency in metamodel definition, and in the broader context to contribute to AMME. The approach introduces the concept of reusable metamodel fragments, accompanied by a comprehensive set of metamodel composition operators for combining these fragments. By leveraging reuse, such a modular approach allows for systematic, flexible, and efficient definition and customization of metamodels.

This paper provides an overview of modular metamodel engineering. Additionally, it delves into potential applications of microservices in the context of modeling method microservices architecture and discusses the potential application of generative AI, shedding light on its implications for advancing metamodeling practices. Following the introductory motivation, Sect. 2 provides a comprehensive overview of the modular metamodel engineering approach. Section 3 discusses how modular metamodeling can benefit from microservice architecture and generative AI. Finally, Sect. 4 concludes the work.

2 Modular Metamodel Engineering

The Modular Metamodel Engineering (MME) approach may be seen as a continuation of the fragment-based method integration idea proposed in [10]. Focusing on the language part of the method, and on the metamodel as a pivotal element in language definition, MME provides a systematic formalism for the realization of modular metamodels within metamodeling platforms. MME aims at extending the core metamodeling concepts, i.e. the concepts of a metamodeling language, to allow for the modular metamodel definition. On the one side, it introduces concepts to systematically define reusable, self-contained metamodel fragments. On the other side, it aims at extending metamodeling languages with a set of composition operators to holistically support white-box, grey-box and black-box composition. Figure 1 illustrates the anatomy of the approach and its main contributions in metamodeling. While core metamodeling refers to the basic metamodeling concepts for constructing metamodels, i.e. for "metamodeling-in-

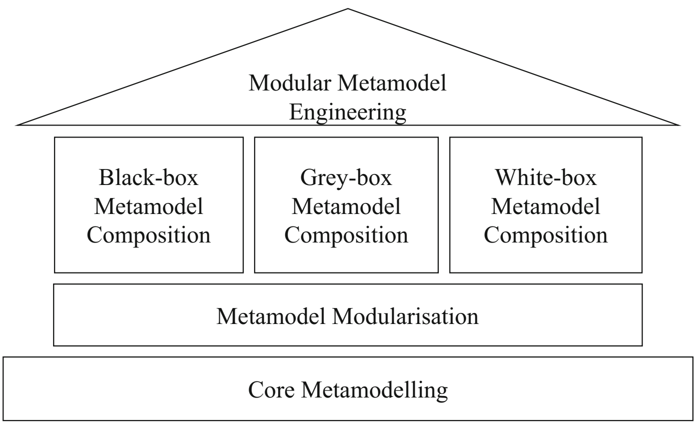


Fig. 1 Pillars of Modular Metamodel Engineering

the-small", metamodel modularization contributes to that core with concepts for encapsulation and information hiding in metamodeling. On top of modularization concepts, comprehensive metamodel composition operators for black-box, grey-box and white-box composition allow for flexible combination of metamodel fragments to support the idea of "metamodeling-in-the-large". In the following, each of these building blocks is introduced.

2.1 Core Metamodeling Concepts

Core metamodeling concepts allow for defining the core metamodel structure. A multitude of mature metamodeling languages exist that formalize those concepts such as the standard MOF [11], or tool-specific meta-languages such as Eclipse EMF Ecore [12], MetaEdit+ GOPPRR [13], ADOxx Meta^2-Model [14–16], or GME MetaGME [22]. The fundamental metamodeling concepts that are typical and prevalent in many metamodeling languages include Model Type, Class, Relation, Relation End, and Attribute. This alignment is unsurprising as metamodels, rooted in graph theory, closely correspond to the foundational elements of attributed typed graphs, encompassing concepts like graph, vertex, edge, and attribute. Figure 2 summarizes the core metamodeling concepts. A metamodel consists of metamodel elements, which may either be attributable elements or attributes. An attributable element contains attributes and supports inheritance to facilitate specialization and reuse of attributes within the hierarchies of the same type of element. An *Attribute* denotes a property of a metamodel element and is of a specific attribute type, though

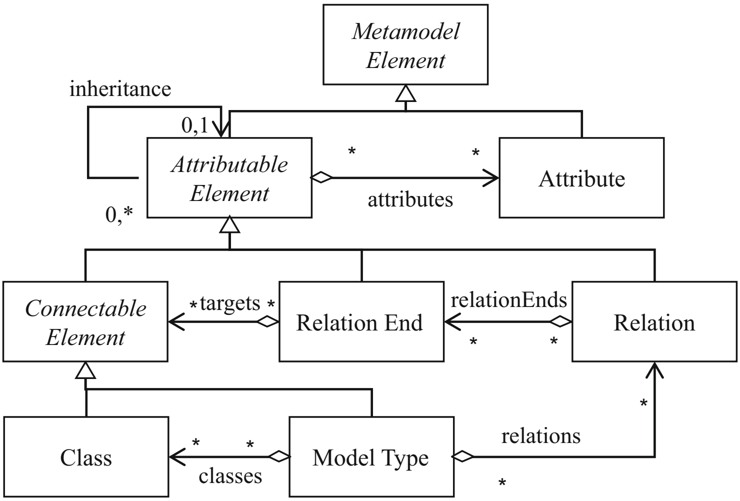


Fig. 2 Core Metamodeling Concepts

we simplify discussion by not further exploring attribute types. A *Class* serves as an attributable element and is a central construct in metamodeling for specifying entities in a modeling language. Another attributable element is a *Model Type*, a meta-construct used to characterize models (diagram types), which itself may encompass classes and relations. A *Relation*, an attributable construct, links classes and/or model types, connecting to other elements indirectly through the concept of a *Relation End*. The arity of the relation is defined by the number of relation ends, and for a relation to be directed, it must have at least one From-type relation end and one To-type relation end. A relation end specifies how a target of a relation participates in the relation, encompassing aspects like multiplicity and direction type. Lastly, a connectable element acts as a facilitating construct to generalize classes and model types as types eligible to be targets of relation ends.

2.2 Metamodel Modularization Concepts

Building upon the definition of a software component [3], *a metamodel fragment is a compositional unit with contractually specified provided and required interfaces. This fragment can be independently deployed and is open to composition by third parties.* This definition introduces key concepts of metamodel modularization, the *metamodel fragment* itself and the *interfaces,* thereby extending the core metamodeling concepts with two crucial modularization capabilities, *encapsulation* and *information hiding.*

Metamodel Fragment Serving as a unit for composition and reuse, a Metamodel Fragment encapsulates metamodel elements contributing to either the implementation (concrete elements) or interface definition (interface elements) of the fragment. The implementation specifies the actual metamodel structure, comprising core elements like model types, classes, relations, relation ends and attributes. Fragments may internally nest other fragments, and elements within a nested fragment are indirectly and recursively owned by the enclosing fragment. A fragment is termed atomic if it solely consists of direct elements, and composite if it includes other fragments.

Explicit and Implicit Interfaces To facilitate information hiding, a fragment defines *explicit interfaces* concealing its internal implementation, thus forming the basis for flexible black-box metamodel composition. *Provided interfaces* expose internal metamodel element implementation, while *required interfaces* specify context dependencies implemented by other fragments. Both interface types may be either *owned* or *imported* from other fragments. An owned required interface represents an extension point of a fragment, i.e. a contract other fragments need to fulfill by wiring to that interface. Fragments with owned required interfaces are abstract. Fragment elements may be available to the outside via interfaces (*black-boxes*) or by controlling their visibility using access modifiers (*white-boxes*), such as public, private, protected etc. Concrete elements that are exposed to the outside via

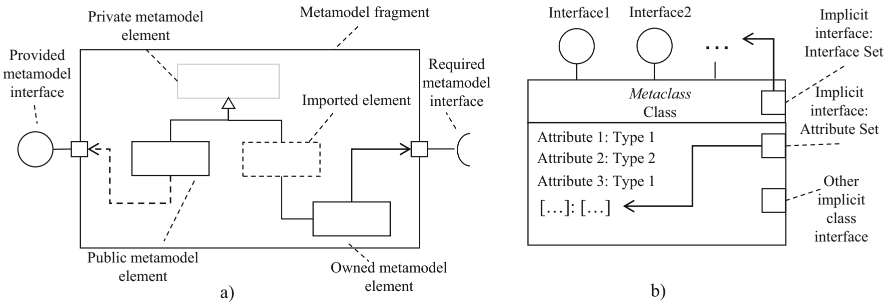


Fig. 3 (**a**) The notion of metamodel fragment. (**b**) The notion of implicit interface based on the Class metaclass

access modifiers can take part in the white-box metamodel composition. The notion of an *implicit interface* is a special form of an interface, which allows for additional extension points of metamodel fragments. An implicit interface, as defined in [8], represents an extension point of a metamodel element, which is implicitly defined by the inherent semantics of the underlying metamodeling language. For example, a set of attributes is an inherent implicit interface of a class, which can be used to inject additional attributes. Fragments that allow for implicit interfaces are called *grey-boxes* and form the basis for the grey-box metamodel composition. Figure 3 illustrates the notion of a metamodel fragment with the idea of explicit and implicit interfaces and access modifiers.

2.3 Metamodel Composition Concepts

Depending on the modularization type (black-box, grey-box, or white-box) of metamodel fragments to be combined, three distinct types of metamodel composition can be identified. The composition of black-box metamodel fragments with explicit interfaces is referred to as black-box metamodel composition. Grey-box metamodel composition involves combining metamodel fragments based on implicit interfaces. Similarly, white-box metamodel composition is applicable to accessible elements within white-box fragments. Each composition type requires suitable metamodel composition operators. Subsequently, the three types of metamodel composition are discussed.

Black-box metamodel composition The black-box metamodel composition involves combining black-box fragments through *explicit interfaces*. To facilitate black-box composition, two interface-based metamodel composition operators are introduced: interface realization and interface subtyping [7].

Interface realization Interface realization binds an internal, concrete metamodel element to an interface element, capturing the realization of an interface by a

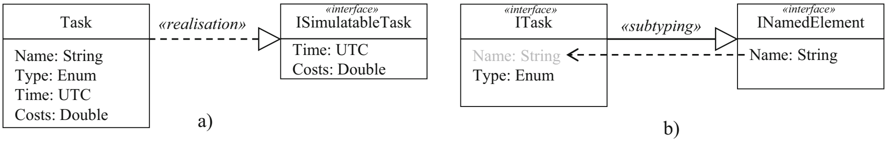


Fig. 4 (**a**) The notion of the interface realization composition operator. (**b**) The notion of the interface subtyping composition operator

concrete element. This realization is valid only if an element conforms to the interface by contract. A concrete metamodel element can provide realizations to multiple interfaces, and a single interface may have realizations from various concrete elements. Interface realization is a fundamental concept in black-box metamodel definition, promoting explicit and controlled exposure of metamodel elements via provided interfaces and the realization of required interfaces within a fragment. For instance, given the interface *ISimulatableTask*, which requires each implementing element to contain attributes *Time* and *Costs* for simulating activities, the class *Task* realizes the interface by providing the required attributes (Fig. 4a).

Interface subtyping Interface subtyping establishes a connection between a base interface and a derived sub-interface, to define substitutability of the base interface by its subtype. The subtype interface can extend the base interface with additional members. Unlike class inheritance, interface subtyping solely inherits member definitions without inheriting their implementations. Furthermore, an interface can extend and be extended by multiple other interfaces. With the ability to extend existing interfaces and declare compatibility between a subtype and its supertype, interface subtyping facilitates highly flexible metamodel composition scenarios at the level of black-box fragments. In Fig. 4b, an abstract *INamedElement* interface requires that each realizing element must include the attribute *Name*. The *ITask* interface can reuse this specification by subtyping the *INamedElement*.

Grey-box metamodel composition The grey-box metamodel composition relies on *implicit interfaces* to combine metamodel fragments into new composites. To compose fragments, the *extension composition operator* comes into play, which adds features to an existing element without modifying it relying on implicit interfaces [8]. This process requires a *base element*, an *extender element*, and an extension composition operator. Extension operator is a relation that takes a base element and an extender element as input and extends the base element by injecting extensions based on implicit interfaces or extension points given by the inherent semantics of the underlying metamodeling language. In the example in Fig. 5a, the base element, the class *Task*, is extended with the *Time* and *Costs* attributes without any syntactic modification of that fragment. The *SimulationActivityExtender* class serves as the extender element containing these attributes. By applying the extension operator on the *attribute set*, as the implicit interface, the class *Task* receives all structural features of the extender *SimulationActivityExtender*, namely the attributes *Time* and *Costs*.

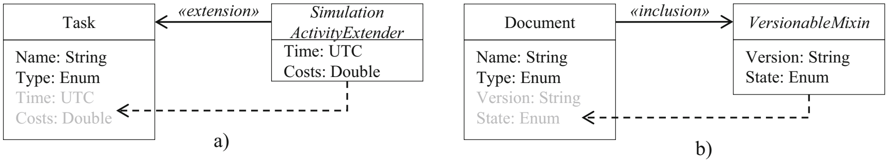


Fig. 5 (**a**) The notion of the extension composition operator. (**b**) The notion of the mixin inclusion composition operator

White-box metamodel composition The white-box metamodel composition combines white-boxes, i.e. accessible internal concrete elements of metamodel fragments using white-box composition operators. If designed carefully and systematically, white-box composition contributes to the systematic decomposition of metamodels into modularized implementation fragments, thereby contributing to fragment interface implementation. Recombining these implementation fragments allows the development of new composite implementations in a productive, modular manner, promoting reuse. As the composition occurs on concrete metamodel elements, standard composition operators like merge, inheritance, and aggregation can be employed. Moreover, referring to the notion of mixins from programming languages, *mixin inclusion* serves as an additional metamodel composition operator that complements standard operators like inheritance and aggregation, thereby enhancing the overall compositional expressivity [8]. The mixin inclusion composition operator accepts a parent element and a mixin element as input, incorporating ("mixing in") the features of the mixin element into the parent. Mixin inclusion is non-invasive, with the parent element serving as the composer, while the mixin remains unmodified as the base element of the composition function. Figure 5b illustrates the application of the mixin inclusion operator. The abstract mixin class, *VersionableMixin*, offers attributes *Version* and *State*. The parent class, *Document*, declares mixin inclusion for the purpose of document versioning and release workflow modules. The application of mixin inclusion from *Document* to *VersionableMixin* results in the *Document* class inheriting all structural features of the mixin class. Finally, the mixin inclusion operator helps in preventing complex inheritance hierarchies in metamodels while enabling inclusion of bundles of structural features defined within mixin elements.

2.4 A Metamodel for Modular Metamodel Engineering

Previously introduced modularization and composition concepts are summarized within a metamodel for modular metamodel engineering (MME). The metamodel serves as a conceptual framework for MME, with the central concept being a *metamodel fragment*. A fragment may contain nested fragments and is categorized as a *composite* fragment if it nests others; otherwise, it is considered *atomic*. Fragments can also declare *dependencies* on each other. A fragment may either existentially *own* metamodel elements or *import* elements from other dependent fragments. Each metamodel element within a fragment, whether it be a class, model type, attribute, relation etc., may represent either a *concrete element* or an *interface element*. Elements are exposed to other fragments via access modifiers. A concrete element may *realize* multiple interfaces, and, conversely, an interface may be realized by any number of concrete elements. With interfacing concept, a fragment having concrete elements may expose a set of provided and required interfaces to other fragments for black-box composition. Other fragments can import, and use provided interfaces or import required interfaces and provide an appropriate realization. In addition to the interface realization composition operation, various other composition operators exist. Interfaces support *subtyping*, allowing easy extension of interface specifications. To enable grey-box composition, the *extension* composition operator is applied. Concrete elements which extend other elements are termed *extender elements*. Fundamental operations for white-box composition, such as *aggregation*, are applicable to both concrete elements and interface elements. On the other hand, *inheritance* and the *mixin* inclusion operator are only applicable to concrete elements. Concrete elements that are included or mixed in by other elements are referred to as *mixin elements* (Fig. 6).

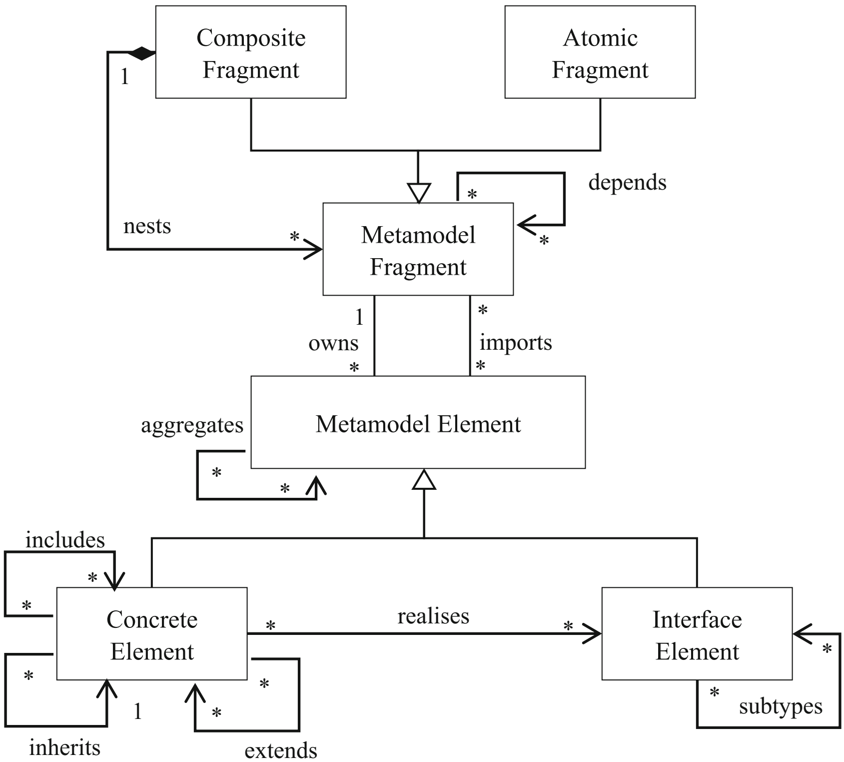


Fig. 6 Modular Metamodel Engineering Concepts

2.5 Implementing Modular Metamodel Engineering

Modular Metamodel Engineering concepts are suitable for implementation within metamodeling environments, which serve as powerful development environments for metamodeling. In [9], the concepts of MME have been formalized in a language for modular metamodel engineering (MMEL). A possible realization of the MMEL has been elaborated based on the metamodeling platform ADOxx [16, 17]. Nevertheless, the MMEL as a meta-language has been designed as a generic modularization capability extension, that could be applied to other metamodeling languages as well.

The application of MMEL for constructing modular hybrid modeling languages within the domain of enterprise modeling, such as Business Process Management (BPM), has been explored in [9]. Specifically, it has been demonstrated how the metamodel of the Business Process Modeling Systems (BPMS) method [18], originally conceived as a monolithic design artifact, can be transformed into a modular implementation through the application of MMEL. The BPMS method, a robust hybrid modeling approach widely employed for enterprise-wide business modeling serves as the central modeling formalism in the business process modeling tool ADONIS [19]. As a hybrid DSML, BPMS includes the process modeling standard BPMN 2.0 [1] and extends it with DSMLs for goals, documents and products, organization, IT modeling, and risk and controls.

Figure 7 illustrates an excerpt of the MMEL usage for the purpose of modularizing the BPMS modeling method, focusing on the interface-based black-box metamodel composition of metamodel fragments, *Business Process Diagram*, *Organization Model* and *Risk Catalogue* using the MMEL notation. The metamodel fragment *Business Process Diagram* is illustrated as a black-box fragment with two major supported interfaces, model type interface *IProcessDiagram* and class

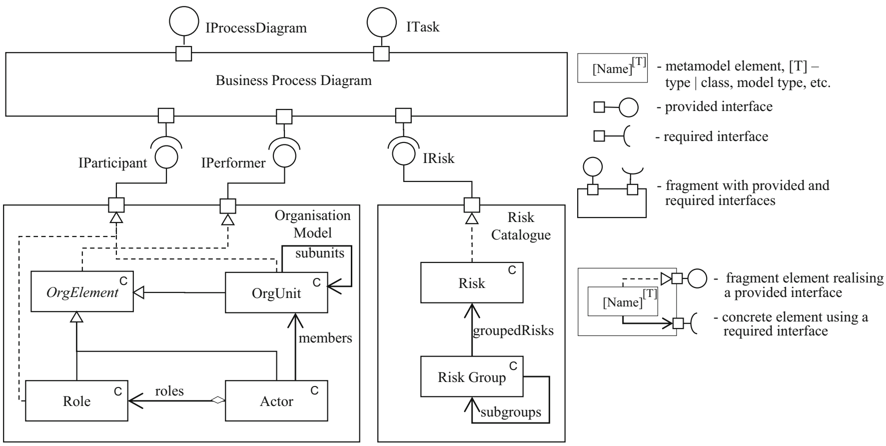


Fig. 7 Modularization of the BPMS Method using MMEL

interface *ITask*. On the other side, required class interfaces *IParticipant*, *IPerformer* and *IRisk* have been defined as extension points for arbitrary fragments to provide appropriate implementations. This way, well-defined interfaces between BPMN and related domains of organization modeling and risk management are established, keeping the domains clearly separated. The composition with fragments *Organization Model* and *Risk Catalogue* is specified based on the *interface realization* composition operator. Concrete classes *OrgElement* and *Role* implement the imported interface *IParticipant*. The interface *IPerformer*, representing task performer role, is realized by concrete classes *OrgUnit*, *Role* and *Actor*. Finally, the BPMN and *Risk Catalogue* fragments are combined by specifying the interface realization between the class *Risk* and the class interface *IRisk*.

3 What's Next

Among the various technological trends that have surfaced in recent years, microservice architecture and artificial intelligence stand out as having significant relevance to the further advancement of modular metamodel development and metamodeling in general. This section delves into these trends and discusses their potential applications.

3.1 MME and Microservice Architecture

In recent years, microservice architecture has become the preferred software design approach for constructing modular and scalable systems, by combining small, independent, and loosely coupled services [20]. It structures an application as a collection of small, independent services, where each service is focused on a specific business capability, and can be developed, deployed, and scaled independently. Microservices communicate with each other through well-defined interfaces (APIs), typically over a network. The microservice approach relies on the same modularization principles that form the core of component-oriented design, but applied at the level of web services, emphasizing the deployment and operational aspects of the service.

Modeling Method Microservices Metamodel fragments are suitable to be realized as microservices due to their modular characteristics. Moreover, positioning it within the broader context of modeling method engineering, and in addition to metamodel fragments as microservices, other modeling method elements as defined in [6] can be realized as *modeling method microservices*. Such modeling method microservice can internally be composed out of metamodel and/or functional microservices. Metamodel fragment interface specifications serve thereby as contracts not only between metamodel fragments but also between metamodel and

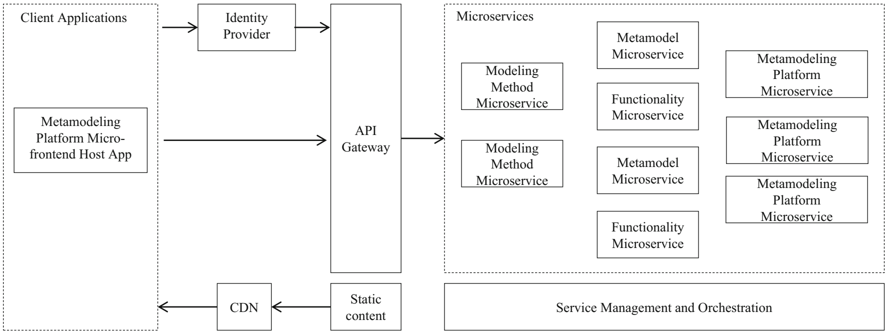


Fig. 8 Modeling Method Microservices Architecture

functionality services. Packaged around modeling business capabilities, modeling method microservices can be independently developed, deployed, scaled, and combined to build a modeling solution addressing the evolving modeling requirements. Figure 8 illustrates the idea of modeling method microservices based on the general microservice architecture.

Benefits and Challenges The significant advantages of adopting the microservice approach for MME, in addition to the already established benefits of modularity, are embedded in the *deployment* and *operational* aspects. This becomes evident in the realization of MME within metamodeling environments.

- *Technological and Platform Independence*. Microservices allow for the use of diverse technologies for individual services, tailoring technology choices based on specific requirements. This flexibility empowers the adoption of the most suitable tools and frameworks for each microservice implementation.
- *Autonomous and Independent Deployment*. Services can be deployed independently, due to the small size of individual services. This means that any modifications made to a modeling method microservice covering one business capability won't impact the entire modeling solution. Even the versions and customization variants of the same service can be deployed and run independently, thereby introducing the additional autonomy and clear separation of concerns between standard and customized modeling method solutions. Furthermore, such microservices support continuous integration and deployment practices, enabling rapid and frequent releases. This is essential for staying in pace with dynamic modeling requirements.
- *Scalability and Resilience*. Microservices enable horizontal scalability. Modeling method microservices can be scaled independently based on their demand. This flexibility is crucial for handling varying workloads efficiently. Due to their independent nature, microservices can be more resilient to failures. If one service experiences issues, it doesn't necessarily affect the entire modeling solution, as other services can continue to function.

- *Service Granularity as a Challenge*. Finding the appropriate service granularity is an important consideration and challenge when adopting microservice architectures. Opposite to a monolithic system, a system with too many fine-grained microservices may lead to increased communication overhead between services, and increased overhead in managing and operating those services. Organizing modeling method microservices around domain-specific modeling capabilities on the one hand, and on customization packages on the other hand could be a good alternative to start when adopting microservices for modeling method engineering.

3.2 MME and Generative AI

Recent advancements in Generative Artificial Intelligence (GenAI) [23], specifically through the application of Large Language Models (LLM), have introduced new possibilities in the field of software development. These advancements enhance various aspects of the development lifecycle, particularly focusing on automated code generation and related assistive services.

- *Metamodel Code Generation*. Metamodels in general and metamodel fragments in particular are specified using MME languages such as MMEL that are provided by a metamodeling language either using a textual, graphical or any other type of concrete syntax. Hence, just like with any programming language source code, generative AI models can be enriched with the MMEL syntax and trained based on large metamodeling code repositories to understand coding patterns and generate metamodel code snippets or even entire fragment specifications. This can accelerate the metamodel development process, especially in the context of metamodel fragment compositions. For example, Amazon CodeWhisperer [21] is an AI-powered productivity tool for the development environments that generates code suggestions based on comments and existing code.
- *Natural Language Processing for Metamodel Specifications*. GenAI, particularly natural language processing (NLP) based on LLMs, can be trained to translate natural language domain-specific metamodel specifications into metamodeling language specifications and generate code for fragment definitions and compositions. Again, MMEL as a formalized language for MME can serve as a knowledge base to train the LLM. This approach has the potential to enhance the accessibility and adoption of metamodeling environments as no-code platforms for non-technical domain expert business users, enabling them to seamlessly take on the roles of both producers and consumers of modeling solutions.

4 Conclusion

This paper provided an overview of Modular Metamodel Engineering as a systematic approach to metamodeling, aiming to enhance the flexibility and efficiency in metamodel definition. The cornerstone of the approach is the concept of reusable metamodel fragments and a set of comprehensive metamodel composition operators, focusing on black-box, grey-box, and white-box composition. The concepts of MME have been implemented in a language for modular metamodel engineering (MMEL), which forms the foundation for introducing MME within metamodelling environments. A brief demonstration of MMEL was presented, illustrating how MME can transform a monolithic BPMS metamodel into a modular specification. Looking ahead, the paper explored the intersection of MME with two emerging technological trends—microservice architecture and GenAI. It envisions modeling method microservices, comprising both metamodel and/or functional microservices, as units of modularization and composition for service-oriented metamodeling environments. Additionally, the paper envisions the role of GenAI in metamodel code generation and NLP for metamodel specifications, offering innovative solutions for accelerating metamodel development and making metamodeling environments accessible to non-technical business experts. Finally, Modular Metamodel Engineering emerges as a promising paradigm for advancing modeling method engineering, providing a structured and adaptive approach to metamodel definition and composition. Placed in the context of agile modeling method engineering frameworks such as AMME, MME contributes to the future where modeling solutions are not only robust but also highly responsive to the dynamic demands of the ever-evolving modeling requirements.

References

1. Object Management Group: Business Process Model and Notation (BPMN) Version 2.0.2, http://www.omg.org/spec/BPMN/2.0.2/PDF (2013)
2. Karagiannis, D.: Agile modeling method engineering. PCI ’15 - Proceedings of the 19th Panhellenic Conference on Informatics, October 2015, pp. 5–10. ACM Digital Library (2015)
3. Szyperski, C.: Component Software: Beyond Object-Oriented Programming, 2nd edn. Addison-Wesley Longman, Boston, MA (2002)
4. Wikipedia, Microservices. https://en.wikipedia.org/wiki/Microservices, last accessed 2024/1/14
5. Selic, B.: The Theory and practice of modeling language design for model-based software engineering - a personal perspective. In: Generative and Transformational Techniques in Software Engineering III, pp. 290–321. Springer (2011)
6. Karagiannis, D., Kühn, H.: Metamodelling platforms. Invited paper. In: Bauknecht, K., Tjoa, A Min., Quirchmayer, G. (Eds.) Proceedings of the Third International Conference ECWeb 2002 - Dexa 2002, Aix-en-Provence, France, September 2–6, 2002, LNCS 2455, Springer, Berlin, Heidelberg, p. 182 (2002)
7. Zivkovic, S., Karagiannis, D.: Towards Metamodelling-in-the-Large: Interface-based Composition for Modular Metamodel Development. In: Enterprise, Business-Process and Information Systems Modeling, pp. 413–428. Springer (2015)

8. Zivkovic, S., Karagiannis, D.: Mixins and extenders for modular metamodel customisation. In Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1, pp. 259–270 (2016)
9. Zivkovic, S.: Metamodel Composition for Hybrid Modelling: A Modular Approach. PhD Thesis, University of Vienna, Vienna (2016)
10. Kühn, H.: Methodenintegration im Business Engineering (English: Method Integration in Business Engineering). PhD Thesis, University of Vienna, Vienna (2004)
11. Object Management Group: Meta Object Facility (MOF) Version 2.4.2: http://www.omg.org/spec/MOF/2.4.2/ (2014)
12. Steinberg, D., Budinsky, F., Merks, E., Paternostro, M.: EMF: Eclipse Modeling Framework. Pearson Education (2008)
13. Kelly, S., Lyytinen, K., Rossi, M.: MetaEdit+ a Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment. In: Advanced Information Systems Engineering, pp. 1–21. Springer (1996)
14. Junginger, S., Kühn, H., Strobl, R., Karagiannis, D.: Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation - ADONIS: Konzeption und Anwendungen. Wirtschaftsinformatik. **42**(5), 392–401 (2000)
15. Kühn, H.: The ADOxx Metamodelling Platform. In Workshop on Methods as Plug-Ins for Meta-Modelling, Klagenfurt (2010)
16. ADOxx Metamodeling Platform: https://www.adoxx.org/ (last accessed: 27.01.2024)
17. The OMiLAB Community. (2022). Development of Conceptual Models and Realization of Modelling Tools Within the ADOxx Meta-Modelling Environment: A Living Paper. In: [15]. https://link.springer.com/chapter/10.1007/978-3-030-93547-4_2
18. Bayer, F., Kühn, H.: Prozessmanagement für Experten: Impulse für aktuelle und wiederkehrende Themen. In German, Springer DE (2013)
19. BOC Group, ADONIS: https://www.boc-group.com/en/adonis/ (last accessed: 28.01.2024)
20. Wolff, E.: Microservices: flexible software architecture. Addison-Wesley Professional (2016)
21. AWS, Amazon CodeWhisperer, https://aws.amazon.com/codewhisperer/, last accessed 2024/1/28
22. Ledeczi, A., Maroti, M., Bakay, A., Karsai, G., Garrett, J., Thomason, C., Nordstrom, G., Sprinkle, J., and Volgyesi, P.:. The generic modeling environment. In Workshop on Intelligent Signal Processing, Budapest, volume 17 (2001)
23. Wikipedia, Generative artificial intelligence, https://en.wikipedia.org/wiki/Generative_artificial_intelligence, last accessed 2024/1/28