Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem

Dominik Bork

University of Vienna, Research Group Knowledge Engineering, Waehringer Street 29, 1090 Vienna, Austria

dominik.bork@univie.ac.at

Robert Andrei Buchmann

Babes-Bolyai University,

Business Informatics Research Center

Str. T. Mihali 58-60, 400591 Cluj-Napoca

Dimitris Karagiannis

University of Vienna,

Research Group Knowledge Engineering

Waehringer Street 29, 1090 Vienna

Moonkun Lee

Chonbuk National University,

Jeonbuk, Republic of Korea

Elena-Teodora Miron

OMiLAB NPO,

Lützowufer 1, 10785, Berlin

Abstract:

This paper motivates, describes, demonstrates in use, and evaluates the Open Models Laboratory (OMiLAB) - an open digital ecosystem designed to support conceptualization and operationalization of conceptual modeling methods. The OMiLAB ecosystem is motivated by a generalized understanding of "model value" and is targeted to research and education stakeholders fulfilling various roles in a modeling method's lifecycle - modelers, domain experts, methodologists, modeling software developers, knowledge workers, model-driven software engineers etc.

While much is reported on novel modeling methods and tools for various domains, only limited knowledge is available on the conceptualization of such methods by means of a full-fledged dedicated open ecosystem and a methodology that facilitates entry points for novices, as well as an open innovation space for experienced stakeholders. This gap is maintained by the lack of an open process and platform for (a) conducting research in the field of modeling method design, (b) developing agile modeling tools and model-driven digital products, and (c) experimentation with, and dissemination of such methods & related prototypes.

OMiLAB incorporates principles, practices, procedures, tools and services required to address the forecited issues, as it aims to be the operational deployment for a conceptualization and operationalization process built on several pillars: (a) a granularly defined *"Modeling Method"* concept whose building blocks can be customized for the domain of choice; (b) an *"Agile Modeling Method Engineering"* framework supporting quick prototyping of modeling tools; (c) a model-aware *“Digital Product Design Lab”*; and (d) dissemination channels for reaching a global community.

Demonstration and evaluation of the OMiLAB in research is hereby carried out by two selected application cases for domains- and case-specific requirements: the *iStar case* for requirements engineering, and the *EnterKnow case* for semantic business process management systems. These two cases show the broad spectrum of modeling methods realized within the OMiLAB, ranging from conventional conceptual modeling methods (iStar) toward the model-aware development and operationalization of Digital Products (EnterKnow). Besides these exemplary cases, OMiLAB has proven to effectively satisfy requirements raised by almost 50 modeling methods, supporting researchers in designing novel modeling methods, developing tools and disseminating outcomes. The educational impact of the OMiLAB is also measured in terms of international visibility and feedback from the NEMO Summer School series, whose attendants directly interact with both the physical and virtual environments of the OMiLAB.

Keywords: OMiLAB, open models, domain-specific conceptual modeling, model value, Agile Modeling Method Engineering

1 Introduction

In the advent of digital transformation and emerging paradigms like the Internet of Things or Industry 4.0, innovative product-service systems need to be designed to support changes in the way humans employ information technology and engage in processes. These changes come with novel requirements for modeling methods and tools. Modeling standards like BPMN and UML are limited in their ability to cope with the needs of such emerging paradigms for three reasons: first, they aim for general applicability, therefore such standards are designed for a higher level of abstraction, ignoring domain-specific aspects to a large extent; second, the core value of modeling standards is in broad adoption and stability, therefore their update cycles are rather long and updates are mostly incremental; third, they hardly operationalize the knowledge that is codified in the conceptual models. It is therefore one recent focus of business information systems engineering research to explore novel domain-specific conceptual modeling languages (Frank et al., 2014) and to develop modeling tools that enable their efficient application (Brenner et al., 2014). Conceptual modeling has become an established research field in information systems (cf. Recker, 2015). Simultaneously, an increasing amount of interest is invested in openness in business and information systems engineering research (van der Aalst et al., 2016).

The proposal presented in this paper originates from a generalized notion of *"model value"* - and requirements with respect to research and teaching arising from it. The Open Models Laboratory (OMiLAB) is hereby introduced as a digital ecosystem that fosters open innovation in conceptual modeling research and teaching. OMiLAB has accumulated results during several years of operation. At its implementation core, it is an orchestration of various enablers that together deploy a conceptualization and operationalization process for modeling methods. Strategically, it is an open community and resource repository supporting the desideratum of "modeling for the masses" which was recently made explicit and roadmapped by the European enterprise modeling community (Sandkuhl et. al., 2018). By generalizing the idea of model value beyond traditionally established application areas (Business Process Management, Software Engineering etc.), it is also a reflection of the value creation principles formulated in the *Memorandum on design-oriented information systems research* (Österle et al. 2011) – i.e., models and modeling methods are not treated only as fixed artifacts observed through a behavioristic lens, but also as dynamic artifacts subjected to design-related concerns and agility requirements, benefitting from an accumulation of community-sourced practices, design patterns, and learned lessons.

OMiLAB addresses three grand research challenges raised by Becker et al. (2015), i.e., *"supporting effective collaboration and learning through evolving media repertoires"*, *"raising collective consciousness"*, and *"developing model-driven methods and tools for the full-scale automated generation of implementation-ready IS"*. At the same time, it is aware of pitfalls and fallacies with respect to the implementation of open innovation communities (von Briel & Recker, 2017). Following these challenges, the OMiLAB strives at establishing an ecosystem platform that enables prototyping, experimentation, and practice-oriented information science research, e.g., related to method engineering (Bucher et al., 2007). An ancillary goal is to also act as an open community facilitator for conceptual modeling, similarly to how Wikipedia supports general purpose knowledge creation and distribution, or how platforms like Apple iOS and Google Android established mobile apps ecosystems (cf. Benlian et al., 2015). The community shares experience and knowledge assets, raises modeling requirements, and evaluates prototypical solutions.

This vision becomes manifest in the pillars the OMiLAB builds upon: (a) a granularly defined *"Modeling Method"* concept whose building blocks can be customized for specific domains; (b) an *"Agile Modeling Method Engineering"* (AMME) framework supporting quick prototyping of modeling tools; (c) a model-aware *"Digital Product Design Lab"*; and (d) dissemination channels for reaching a global community. These enable the new conceptual modeling research framework devised by (Sabegh & Recker, 2017b), which pushes forward the conceptual modeling agenda along several directions – e.g., novel forms of representation, agile modeling, modeling for other purposes than information systems development.

The remainder of this paper is structured as follows: Section 2 outlines the problem, the derived requirements and summarizes how they have been addressed by the OMiLAB. Section 3 details the conceptualization process that is enabled by the OMiLAB Digital Ecosystem. Thereafter, the open innovation community of the OMiLAB is described in Section 4. Section 5 reports on two modeling method conceptualization application cases. Related approaches are briefly discussed and compared to the OMiLAB in Section 6. A multi-faceted evaluation is presented in Section 7. Eventually, conclusions are drawn and future work is pointed to in Section 8.

2 Problem Statement and Solution Overview

In Information Systems education, conceptual modeling is often taught as subsidiary to other disciplines – e.g., Business Process Management, Database Design, Software Engineering - relying on well-established modeling languages that satisfy some fundamental use cases (process simulation, code generation etc.). The unfortunate effect is a *limited perception of model value* – i.e., conceptual modeling is not perceived as a discipline by itself (Buchmann & Ghiran, 2017); instead, it is fragmented into several techniques attached to other disciplines and subordinated to selected modeling languages. In research, on the other hand, conceptual modeling challenges often extend well beyond those teaching cases, opening towards a diversity of paradigms - Knowledge Management, Enterprise Architecture Management, Design Science etc. In a Design Science context (Hevner et al., 2004), a modeling tool, language, or method may be considered a standalone artifact, it does not have strictly instrumental value as a means to some end defined by software engineering.

Consequently, a gap between the two perspectives becomes inherent. In order to bridge it, researchers and practitioners must be empowered with the ability to conceptualize modeling methods that are themselves artifacts, and are valued with respect to requirements emerging from domain-specific or even case-specific needs. A **conceptualization process** to support this vision, and an **ecosystem** where modeling stakeholders can easily become productive in **operationalizing** it, are therefore necessary. This paper introduces OMiLAB, which is an exemplar of such an ecosystem built around a conceptualization and an operationalization process that was designed for a generalized notion of "model value".

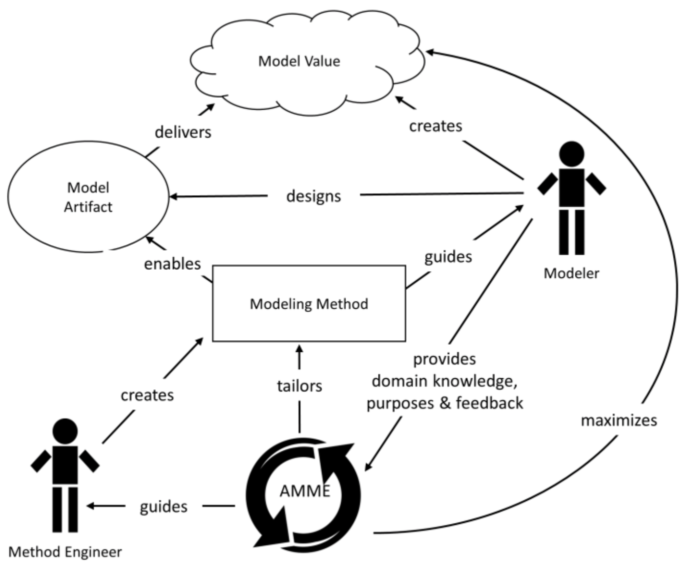

Figure 1. The key OMiLAB motivator: the co-creation of model value

Model value must be detached from a particular usage or modeling language. It needs to encompass both *instrumental value* (as enabler for a model-aware digital product) and *intrinsic value* (as a means of knowledge representation). At the same time, it must be aligned to modern value theories such as that of "value co-creation" established as an economic strategy of stimulating collaboration for mutually valued outcome (Galvagno & Dalli, 2014). However, co-creation implies interactive relationships and at least two contributing roles, as illustrated in Figure 1 in the context of OMiLAB:

1. The *Modeler* is the main source of modeling requirements and responsible for designing model artifacts, therefore having a direct influence on model value. This value is linked to purposes (reflected in modeling requirements) which may evolve (reclaiming agility), and is largely determined by the knowledge that can be acquired while engineering the modeling method. Furthermore, the modeler also acts as the primary source of feedback, thereby steering the evolution and improvement of the modeling method;
2. The *Method Engineer* is guided by the Agile Modeling Method Engineering framework (Karagiannis, 2015) to customize a modeling method according to evolving needs and a gradual understanding of the application domain, thus enabling model value and maximizing it while fine tuning the expressiveness of models and providing adequate guidance to the modeler.

The research question guiding this paper is: *How can we enable the agile co-creation of model value, considering both instrumental and intrinsic criteria?* As solution we propose a conceptualization process and an ecosystem to operationalize this process and the resulting model-aware digital product. OMiLAB is the operational deployment of the conceptualization process proposed to support the co-creation strategy visualized in Figure 1 which, eventually, may become the core principle of an economy of model value chains. The observation of such value chains and the refinement of co-creation models across a multitude of projects is a long-term goal of this digital ecosystem.

This research question was abstracted from the requirements of more than 50 projects hosted by OMiLAB. These requirements are discussed on a more granular level in Table 1, as well as a summary of how they are addressed by various artifacts provided by OMiLAB. These treatments will further be detailed in the subsequent sections.

Table 1: OMiLAB requirements and their respective treatments

Requirement	How OMiLAB addresses the requirement
The **Decomposition requirement.** There is a need to manage the complexity of a modeling method by separating concerns regarding its building blocks. This is closely related to agility, as different components of a modeling method may have to be developed by different teams, so they must be allowed to evolve separately (considering possible dependencies).	*The OMiLAB conceptualization process was defined around the "seed concept" of modeling method as introduced in (Karagiannis & Kuhn, 2002), where its building blocks and dependencies are prescribed in a decompositional manner. The concept was later mapped on features of the metamodeling platform ADOxx (ADOxx.org, 2018) to support formal analysis of such methods (see Fill et al. 2012). Further mappings to other platforms are also possible. One project's experience report (Buchmann & Karagiannis, 2015) discusses different types of requirements propagation among the modeling method's building blocks, thus guiding a granular structuring.*
The **Agility requirement.** There is a need to enable agility in all the building blocks of a modeling method (treated as backlog items). Agility is understood here as responsiveness to changing needs and is a recognized quality in software engineering methods, but is less common in modeling methods. That is, modeling languages are often perceived as invariants with respect to development processes, or as pursuing the ambition of universal consensus and standardization.	*OMiLAB empowers its stakeholders to conceptualize and rapidly prototype modeling methods based on their application case requirements, or propagating from model-aware digital products*. *This agility is enabled by an incremental and iterative framework called Agile Modeling Method Engineering (Karagiannis, 2015). Instead of focusing on researching interoperability mechanisms across existing modeling tools, stakeholders can agilely tailor their own modeling tool, either from scratch or by reusing already implemented building blocks - see the BEE-UP open tool (Karagiannis et al., 2016b).*
The **Knowledge ecosystem requirement.** Co-creation raises a need for a shared repository of knowledge assets, practices and deliverables that can support knowledge acquisition and knowledge transfer among projects, engineering phases, stakeholder types, even application domains.	*OMiLAB provides a portal organized around projects of specific granularity, considering the prescribed building blocks and the typical deliverables that are produced during modeling method conceptualization, including specification documents or evaluation protocols. It also provides the user management mechanisms to enable a social dimension for interactions within its community.*
The **Openness requirement.** Co-creation implies interactive relationships, which are typically found at the core of open innovation communities. Open use and open source implementations, as well as reusable	*Openness is the fundamental motivator and philosophy for OMiLAB (Bork and Miron, 2017) with respect to both, the technological resources used during the conceptualization, and to the knowledge artifacts thereby created.*

services and tutorials, facilitate interaction by lowering entry barriers for novices. At the same time, they provide best practices or reference implementations to domain experts.	
The **Technology requirement.** There is a need for technological enablers that facilitate fast prototyping, considering both the intrinsic value of models (as knowledge assets) and their instrumental value (as input for some model-aware artifacts).	*A key resource of OMiLAB are open use metamodeling enablers, modeling tools and various plug-ins for developing model-aware digital products (e.g., model serialization plug-ins, microservice frameworks for model querying/publishing).* *A platform-independent declarative language for modeling method definitions is being developed (Visic et al., 2015). This suggests that other metamodeling platforms may also become OMiLAB resources, as long as their creators are willing and able to support the foundational philosophy of openness, flexibility, sustainability, and maturity advocated by the OMiLAB.*
The **Dissemination requirement.** In academic and scientific communities, a particular kind of incentive is given to dissemination. There is a need to publish artifacts or experimental results generated during the conceptualization and operationalization of modeling methods.	*The OMiLAB portal and specific services have been tailored as a dissemination channel where method engineers manage their own workspace, structured around the typical artifacts that are built during modeling method conceptualization*. *Additionally, OMiLAB events and joint publications foster a community of innovation, embracing a shared understanding of model value.*

To address these requirements, the conceptualization process underlying the OMiLAB digital ecosystem was established, with its key community roles and building blocks as visualized in Figure 2. Depending on project complexity and requirements, the same actor may fulfil multiple stakeholder roles, or the same role can be shared among multiple contributors:

- *Domain Expert* – the source of domain knowledge that must be acquired and embedded in all building blocks of a modeling method – i.e., in notational specificity (cf. Bork et al., 2018), in concept definitions, in functionality reflecting some aspects of business logic;
- *Metamodel Designer* – responsible for the central building block of a modeling method – i.e., the modeling language definition (covering notation, syntax and semantics);
- *Modeling Method Engineer* – builds the modeling method by extending the language with a modeling procedure and model-driven functionality, and implements them in a usable modeling tool, aiming for qualities such as adaptability, extensibility, usability, operability and integrability; the AMME framework (Karagiannis, 2015) is employed to achieve this output;
- *Digital Product Engineer* – applies the modeling method for a selected domain or case towards the goal of realizing a Digital Product that employs the modeling tool as its "knowledge engine" – i.e., a knowledge acquisition enabler that interoperates with other artifacts (e.g., Internet of Things components, microservices) either through direct interoperability mechanisms or through the model base. Digital Product prototypes that employ models as a knowledge source can thus be built or experimented upon. An m:n relationship exists between the modeling language defined in the first stage and the Digital Products – e.g., a hybridization of modeling languages may be necessary in the same modeling tool – something that is made possible by AMME;
- *The End-user* – experiences the Digital Product prototype which typically includes a modeling tool and software artifacts that interoperate with the modeling environment through some of the available channels (e.g., web service, model serializations). In the absence of a Digital Product, the end-user is strictly a modeler who will use the modeling tool and its customized functionality (e.g., simulation, reporting, model queries).

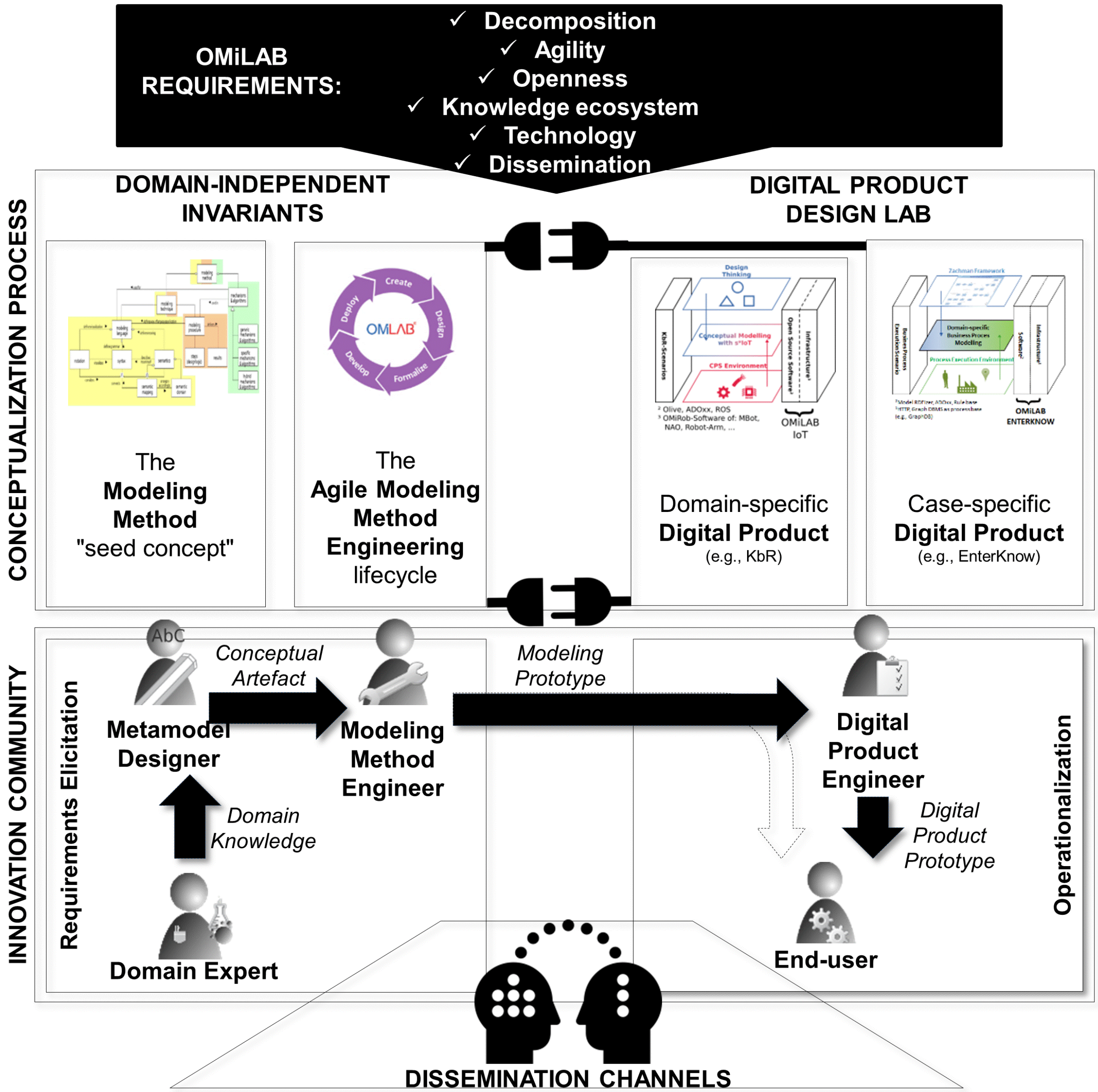

Figure 2. The OMiLAB Digital Ecosystem

All stakeholders of the OMiLAB Digital Ecosystem (cf. Figure 2) are supported with specific enablers, to be detailed in subsequent sections. Each step of the conceptualization process produces artifacts that have a deliverable value (can be disseminated, extended or adapted for other projects). A process-centric view is given in Figure 3, including key OMiLAB resources on the left side and disseminable assets in the right-side.

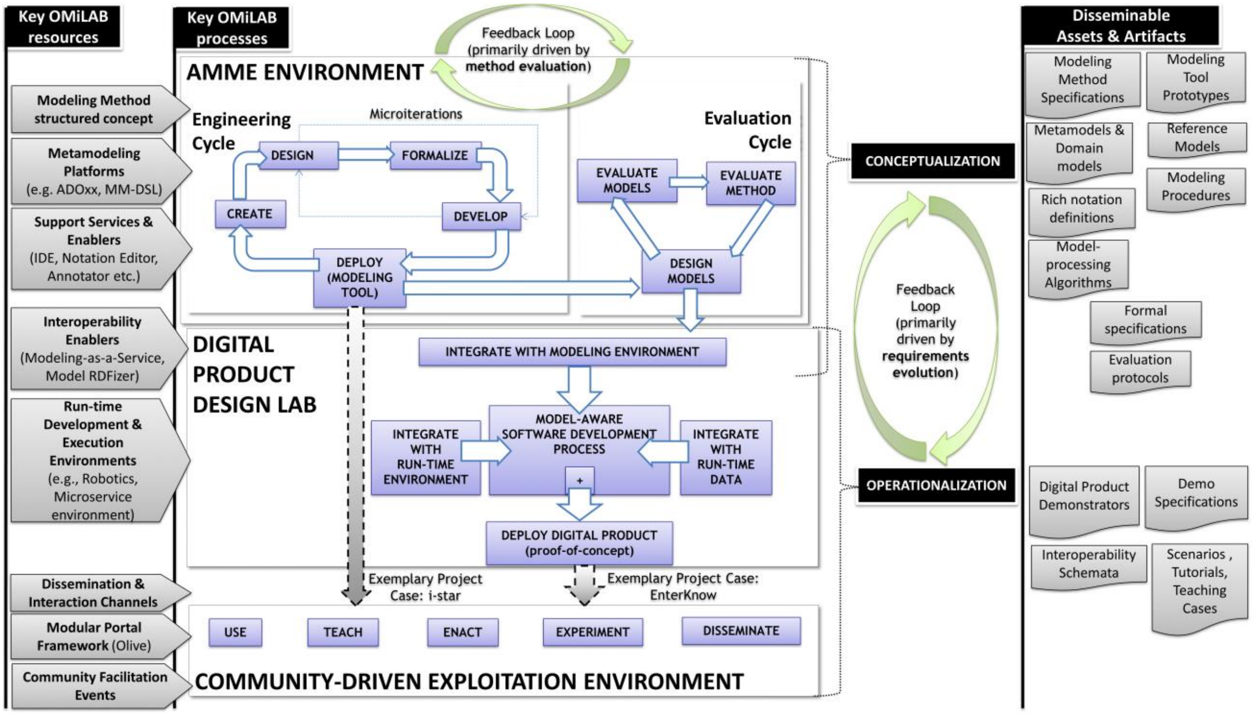

Figure 3. OMiLAB Conceptualization & Operationalization: a process-centric view

The environments that form OMiLAB and their respective key resources will be inventoried in the subsequent Sections 3 and 4. Two exemplary application cases will then be summarized in Section 5 to illustrate the two most common types of projects hosted by OMiLAB – (i) one where the main outcome is an evolving modeling tool prototype (i*, to be discussed in Section 5.1) and (ii) one where the modeling tool is the knowledge core of some "Digital Product proof-of-concept" (in our selected case a business process management system - EnterKnow, to be discussed in Section 5.2). The second type of project involves domain-specific run-time environments for which interoperability enablers are of essence; also, from a methodological standpoint they rely on a flavor of model-driven software development labelled here as "model-aware development" - its key characteristic is that the developed software uses a model base at run-time, as a complement to its more traditional database/data sources. Such model-awareness is commonly implemented as model queries (Buchmann et al., 2018) or via a service-based architecture (Walch & Karagiannis, 2017). From an operationalization standpoint, both types of projects will generate artifacts that are subjected to the exploitation goals of community members, varying from teaching to practical use or scientific experimentation.

3 Modeling Method Conceptualization in OMiLAB

3.1 The Modeling Method Concept

Conceptual modeling methods facilitate the management of complexity by applying abstraction for a specific purpose. According to Karagiannis and Kühn (2002), a comprehensive modeling method is composed of a modeling language, a modeling procedure, and mechanisms & algorithms (see Figure 4). The central building block of a modeling method is a modeling language which can be further decomposed into the syntactic elements (syntax), their graphical representation (notation), and their meaning (semantics). The modeling procedure describes the steps and results of applying a modeling language by a modeler in order to create valid models. Mechanisms & algorithms define the functionality (e.g., simulation, transformation) that should be implemented in a corresponding modeling tool.

Based on the pragmatics of a modeling method, domain-specific modeling methods can be distinguished from general-purpose ones. The former have the potential to address domain specificity in all building blocks of a modeling method, while the latter aim for comparability, interoperability, reusability across domains and possibly standardization. Agility is another emergent requirement for modeling methods, in the sense of supporting the evolution of a method's building blocks (and tool implementation) in answer to changing modeling needs.

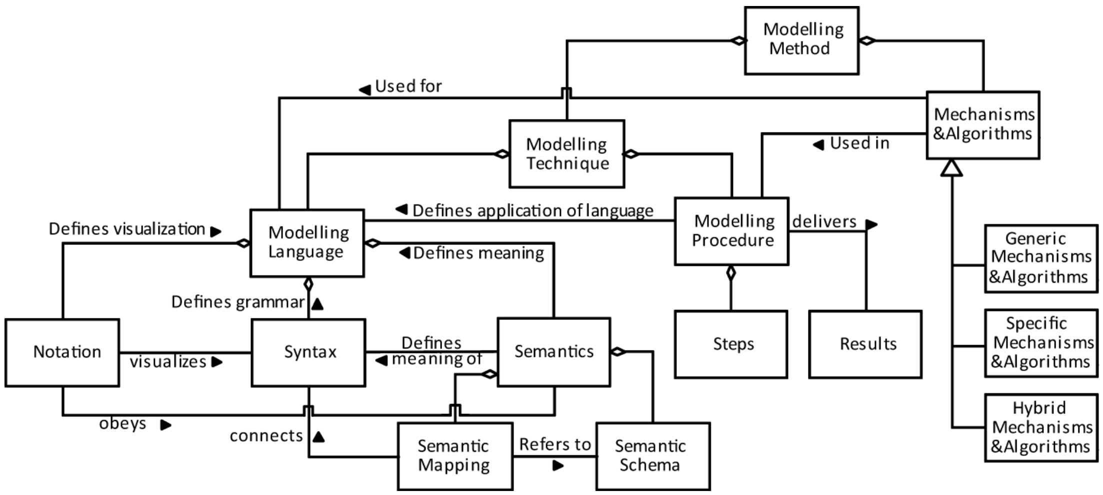

Figure 4. Components of modeling methods (Karagiannis and Kühn, 2002)

A further differentiation can be drawn when considering the application domain of modeling methods. In computer science, modeling methods are designed with the goal of model-driven systems development. Such models often lack proper visualization and focus instead on the capabilities of model transformation and code generation (Moody, 2009). In information science or knowledge management, modeling methods are also used to create abstract representations for "purposes of understanding and communication" (Mylopoulos, 1992) - hence the generalized perspective on model value introduced in Section 2.

3.2 The AMME Lifecycle

The aforementioned "modeling method" concept may be subjected to a requirement of agility – i.e., responsiveness to emergent or evolving needs for extension, adaptation, hybridization or generally customization in every building block of the method. Rapid prototyping enablers must facilitate iterative and incremental implementation, thus moving away from the traditional perception that modeling languages aim for global consensus or standardization. In OMiLAB's generalized understanding of model value, a modeling language may even be customized for a single enterprise, to support some model-aware system or Digital Product. Therefore the structural view provided in the previous subsection must be complemented by a dynamic view considering the specific engineering phases and intermediate artifacts produced during an agile modeling method's lifecycle.

The Agile Modeling Method Engineering (AMME) framework (Karagiannis, 2015) prescribes the iterative lifecycle presented in Figure 3. AMME covers an engineering cycle that starts with knowledge acquisition and requirements analysis - the *create* phase - and ends with the deployment of a usable modeling tool - the *deploy* phase. The intermediate phases are *design*, resulting in a specification of the modeling method building blocks, *formalize*, resulting in a formalism-oriented specification, and *develop*, resulting in an implemented modeling tool. Microiterations can be applied between the *design* and *develop* phases to apply quick changes while skipping the other phases. Each phase is supported by specific methodologies, tools, and best practices hosted by the OMiLAB portal (see Section 4.3). The most prominent resource is the ADOxx metamodeling platform (ADOxx.org, 2018; Efendioglu et al., 2016) providing built-in functionality (e.g., model management, user interaction) that allows method engineers to focus on the modeling method's building blocks without being distracted by generic programming concerns. Consequently, productivity is significantly improved, as developers will spend most of their effort on the following tasks (cf. Bork and Sinz, 2010): 1) metamodel definition by instantiating meta-meta concepts of ADOxx; 2) definition of visualization and linking of concepts across language partitions (model types); and 3) scripting of functionality (model transformations, model queries, simulations etc.).

Agility is enabled by the iterative and incremental nature of the AMME engineering cycle, as modeling requirements are expected to evolve – similarly to how agile software development principles emerged from a need for responsiveness and manageable granularity in software engineering. A feedback loop occurs between this engineering cycle and an evaluation cycle based on hands-on experience with the deployed modeling prototype, as suggested in Figure 3.

3.3 The Digital Product Design Lab

The majority of projects currently hosted by OMiLAB advance up to the deployment of a modeling tool, commonly including some model-aware functionality to maximize model value in various decision-support contexts (e.g., simulation, transformation, model queries). However, some projects advance further to the last phase depicted in Figure 2: the engineering of model-aware Digital Products benefitting from resources of the Digital Product Design Lab component of OMiLAB – typically they are components derived from past projects that have been generalized for reusability, see the microservice architecture for interacting with robots (Walch & Karagiannis, 2017), a plug-in for serialization of models as RDF knowledge graphs (Karagiannis & Buchmann, 2018) or a "model base-as-a-service" architecture provided by ADOxx. The modeling method that is developed through AMME may take its increment requirements (and also modeling concepts) from the elements of a Digital Product Design Framework in which the modeling method acts as a "knowledge engine", on the conceptual modeling layer as depicted in Figure 5.

Modeling method conceptualization is consequently determined by two context facets: (1) the *Application Domain* (through the "business layer") where some high-level conceptual framework (e.g., Design Thinking, Zachman Framework) typically provides an initial set of concepts to be refined through the AMME lifecycle; (2) the *IT Deployment* (through the "proof-of-concept layer") whose technology-specific constructs must be mapped to modeling concepts, possibly imposing constraints and the need for interoperability between the modeling environment and the Digital Product's run-time environment.

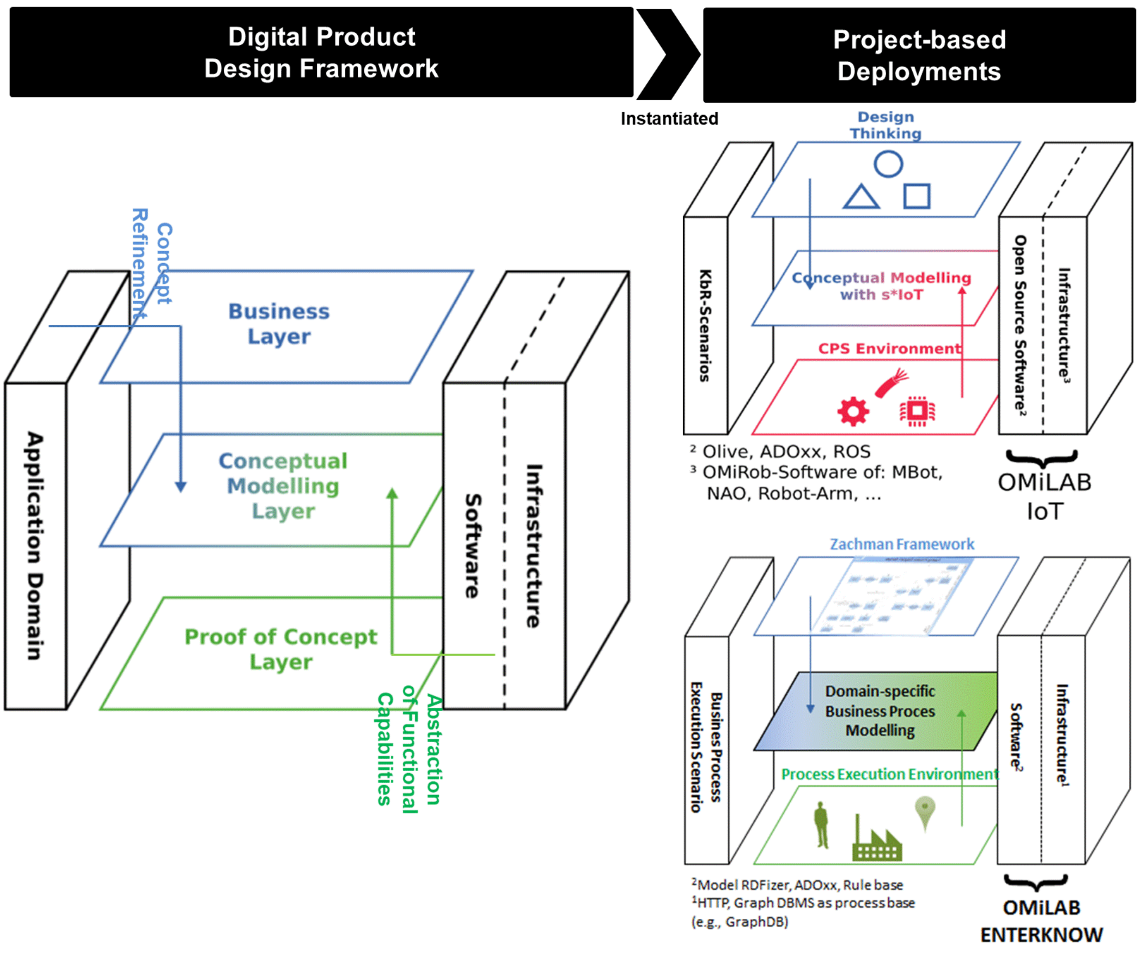

Figure 5. The Digital Product Design Framework

Concepts and properties that are relevant to be assimilated in the agile modeling language may emerge from the "business layer" by typically applying specialization and decomposition, or from the "proof-of-concept layer" by abstracting some functional capabilities of the underlying implementation environment. Some generalized interoperability mechanisms are available from past project instances developed within OMiLAB, as suggested in Figure 5 – e.g. the plug-in for exporting any kind of model to RDF knowledge graphs. The examples indicated in Figure 5 are:

- KbR - a Knowledge-based Robotics project that demonstrates the coordination and operation of robotic arms through diagrammatic models that prescribe behavior, environmental layout, artifacts, and rules (Walch & Karagiannis, 2017; Walch & Karagiannis, 2019). The deployment employs Design Thinking as high-level methodology and Cyber-Physical Systems as a low-level execution context, with an IoT-specific modeling tool establishing the conceptual bridge across these layers, and a microservice-based interoperability framework enabling execution;
- The Enterknow project for which a semantic Business Process Management System demonstrator is developed starting from the abstractions of the Zachman Framework (Zachman, 1987) which are specialized and exposed as machine-readable knowledge graphs to a semantics-driven system to support task execution. Further details on this second case will be presented in Section 5.2 to better highlight how the OMiLAB conceptualization process is extended towards developing model-aware Digital Product prototypes.

4 The OMiLAB Open Innovation Community

The concept of the OMiLAB was inspired by the open source movement and motivated by the belief that the conceptual modeling community is yet to harness its collaborative innovation potential, which was foreshadowed by the notion of "reference models" (Koch et al., 2006). Successful communities tend to center around a technological and dissemination platform, which enables the collective and individuals to contribute their ideas – an initial feasibility study (Karagiannis et al., 2008) assessed the prevalence of modeling editors and tools as well as the need for an open innovation community in conceptual modeling.

Founded in 2008, the OMiLAB community has continually evolved. Pioneering projects were the Business Engineering Navigator (Winter 2010), the Semantic Object Model (Bork & Sinz, 2010; Ferstl et al., 2016), and the exemplary Business Process Modeler (Breitling & Hofer, 2012). Currently almost 50 different modeling methods have been successfully conceptualized in OMiLAB - originating heterogeneous domains, e.g., Multi-Perspective Enterprise Modeling (MEMO) (Bock & Frank, 2016), Design Thinking (Hawryszkiewycz & Prackwieser, 2016; Miron et al., 2018), or Knowledge Management (Cairó & Guardati, 2012; Cairó Battistutti & Bork, 2017). Recent conceptualizations target bridging design thinking and conceptual modeling (Miron et al., 2018; Miron et al., 2019) and industrial business process management.(Utz & Falcioni, 2018) A comprehensive overview of successful conceptualizations is given in Karagiannis et al. (2016). A full project repository is available online at http://austria.omilab.org.

4.1 Collaboration Possibilities

Besides the casual visitors of the OMiLAB portal (see Section 4.3), interested in downloading a modeling tool for their own needs without any other interaction, a structured way of collaboration with the OMiLAB prescribes three levels of commitment: *Member*, *Associated Partner*, and *Operator* (of an OMiLAB node).

Members (mostly individuals) contribute with their domain knowledge and modeling method to the OMiLAB, being incentivized by their need to have a modeling prototype to support experimentation for a publication or thesis. They must make available an open modeling tool for their method. Research groups can become *Associated Partners* (AP) of the OMiLAB, implying the additional commitment to contribute to OMiLAB events and to promote the use of OMiLAB resources (e.g., in university courses). Therefore APs form the core forum that drives OMiLAB's international visibility and publicizes the resources (tools, knowledge assets, methods) emerging from the community. Table 2 depicts the level of adoption of OMiLAB resources in the exemplary case of an university where a Business Information Systems study program employs OMiLAB resources for various topics across the provided study levels.

Finally, the *Operator* status implies the willingness to establish the physical infrastructure and knowledge required to perform trainings and to host OMiLAB resources. Currently two physical OMiLABs exist that ensure regional collaboration facilities - one is located at the University of Vienna, Austria, the other at the University of Chonbuk, Korea. Both labs aim at becoming the primary address for method engineers and tool developers in their respective region. A third OMiLAB in Europe is currently in the final preparation phase. Such distributed physical laboratories will also allow to separate regulatory concerns across different geographical regions, enabling agility in the administrative and managerial aspects of OMiLAB.

OMiLAB community results as well as the international distribution of its collaborators are documented in the wide variety of freely available modeling tools presented in (Karagiannis et al., 2016), a first volume of a joint publication series from the OMiLAB community. Periodic community events are also essential in fostering collaboration and joint publications. Their history[1] covers events with both educational and research purposes in the form of an annual summer school and thematic workshops affiliated to highly visible conferences, respectively.

Table 2: Exemplary adoption of the OMiLAB in a partner university

	TOPICS	OMiLAB resources
Levels >=4: PhD, PostDoc, Project-based Research	Semantic Technology and Conceptual Modeling (with business administration as application domain)	• OMiLAB as hosting and dissemination environment • AMME as Design Science framework • Metamodeling platforms (e.g., ADOxx) and reusable artifacts • Community-driven domain knowledge & learned lessons
Level 3: Master – advanced topics	Enterprise Modeling, Semantic Business Process Management	• Free modeling tools for 4EM, ArchiMate, domain-specific tools • Intermediate deliverables of modeling method specifications & modeling tool implementations
Level 2: Master – early topics	Business Process Management Systems, Knowledge Management Systems	Free modeling tools for BPMN, EPC, Petri Net (e.g., Bee-Up)
Level 1: Bachelor	Database Design, Software Engineering	Free modeling tools for UML, ER (e.g., Bee-Up)

4.2 OMiLAB in Education

The annual Next-generation Enterprise Modeling (NEMO) Summer School[2], initially founded through the Erasmus+ Strategic Partnership Project OMI-KA2, is one key initiative within the OMiLAB enabling a comprehensive evaluation environment for modeling methods and educational materials. Since 2014, distinguished professors from different continents with varying domain backgrounds present their view on next-generation enterprise modeling to an international cohort of Master and PhD students. The NEMO program comprises theoretical lectures and practical exercise sessions applying conceptual modeling in emerging domains such as Smart Cities (Bork et al., 2015; Bork et al., 2016). NEMO enables the geographically dispersed community to have an annual meeting, to share experiences, and discuss future collaboration opportunities. For the students, it is a unique opportunity to interact with professors and other students coming from different cultural and educational backgrounds, as well as to follow highly advanced short-term trainings in various aspects of conceptual modeling.

One key educational resource is the Bee-Up tool[3] which was tailored for teaching requirements to support courses on the foundations of modeling with well-established languages like BPMN, EPC, ER, UML, and Petri Nets. Besides that, OMiLAB community members are encouraged to share training materials for their domain-specific methods, thus enabling testing and experimentation with a broad and diverse audience, leading to the refinement of evaluation protocols that provide valuable input for scientific publications.

Finally, OMiLAB aims to establish valuable training resources for tool developers and digital product engineers in order to facilitate productive prototyping driven by scientific principles – e.g., how to design meaningful and intuitive graphical visualizations (cf. Moody, 2009; Stark et al., 2017; Bork et al., 2018), how to specify metamodels (Bork et al., 2018b), or how to specify mechanisms & algorithms that process the model semantics (Buchmann & Karagiannis, 2016; Buchmann & Karagiannis, 2017).

4.3 The OMiLAB Portal

The OMiLAB Portal is the central space for OMiLAB dissemination and resource hosting. It is openly available at www.omilab.org for the presentation of e.g., modeling projects, events, services and tools. Based on Olive[4], an internally developed content management system deployed in a flexible microservice architecture, each OMiLAB project can disseminate in a structured way information regarding the domain requirements, use cases, tutorials, training materials, a wiki, and download pages for its prototypes, e.g. modeling tools.

[1] OMiLAB Event, http://www.omilab.org/psm/content/ep/globalnetworkservice?view=tilesevents, last visited: 2018-11-29
[2] NEMO Summer School Series homepage, http://nemo.omilab.org, last checked: 2019-11-28
[3] BEE-UP modeling tool homepage, http://austria.omilab.org/psm/content/bee-up/info, last checked: 2018-11-29
[4] OLIVE documentation, https://www.adoxx.org/live/olive, last checked: 2018-11-28

The portal also hosts a variety of tools and services that complement the metamodeling platform. They are typically outcomes of past projects that have been refined to be reusable beyond their original context:

- Integrity of OMiLAB models can be evaluated by the Semcheck project (Jeusfeld, 2016) using the ConceptBase system (Jarke et al., 1995);
- The *GraphRep Generator* helps in designing the notation of the modeling language elements in an intuitive way by providing a What-You-See-Is-What-You-Get drawing editor. Users can draw elements and drag & drop existing composed elements into a drawing area. When the design is finished, ADOxx-specific code can be automatically generated from the drawings;
- MM-DSL (Visic et al., 2015) can be used to specify the constituents of a modeling language by means of a declarative language and then compile the source code to platforms such as ADOxx in order to generate a modeling tool;
- An OMiLAB IDE aims to support the modeling tool development phase of AMME with code completion, debugging and features already known from IDEs like Eclipse;
- Any modeling tool that is developed on ADOxx can export its diagrams to RDF graphs with the help of an external plug-in, introduced in (Karagiannis & Buchmann, 2016) to support (a) interoperability with model-aware digital products and artifacts; and (b) reasoning and semantic queries across multiple related models or model-data mashups.

5 Selected Application Cases

Typical OMiLAB project goals range from (i) producing strictly a modeling tool for specific requirements or scientific experimentation to (ii) implementing a Digital Product that employs such a tool as a knowledge representation instrument. In this Section, the two categories are illustrated by two exemplary selected projects – one tailored for requirements engineering (iStar, presented in Section 5.1) and one showcasing model-aware semantic information systems (EnterKnow, presented in Section 5.2). We refer back to Figure 3 which indicates which path through the overall conceptualization process is taken by each of these projects.

The brief presentations will highlight the project-specific requirements (complementing the high-level requirements discussed in Section 2) as well as their conceptualization as modeling methods within the OMiLAB digital ecosystem. Each case is introduced through the method-specific requirements and means of addressing them.

5.1 The iStar Case

The iStar language initially aimed at modeling strategic relationships for process re-engineering by focusing on intentional, social, and strategic dimensions (Yu, 1995). In recent years, several extensions and adoptions of iStar, model analysis techniques, and modeling tools have been proposed (cf. Horkoff & Yu, 2016; Li et al., 2016). As iStar evolves and is often extended, agility at modeling method level becomes a necessity, therefore making it an ideal case for the AMME framework. In the following, the current conceptualization of iStar within OMiLAB (Franch et al., 2016) is briefly presented[5]. The iStar requirements cover not only notational aspects, but also processing functionality in the modeling environment, such as:

1. *Boundary highlighting* - The relationships between intentional elements and intentional actors need to be intuitively encoded in the notation;
2. *Semi-automated dependency definition* – Dependency creation should be supported intuitively;
3. *Goal Satisfaction Evaluation* - The goal satisfaction shall be computable;
4. *Intentional Actor Relationships* - Modelers should have the possibility to gain an overview of the relationships of one or more intentional actors;
5. *Interpretability* - Strategic dependency model and strategic rational model should be interpretable.

For requirement [1], intentional actors may visualize a boundary to express that all intentional elements within this boundary are explicitly desired by that intentional actor. If no actor is placed in proximity or within a boundary, a hint is shown to the modeler, thereby guiding the creation of semantically valid iStar

[5] Available online at http://austria.omilab.org/psm/content/istar/info, last accessed: 2018-11-29

models. In order to support the definition of dependencies (Requirement [2]), the iStar tool provides a drop-down dialog whenever a new dependency relation has been created. The modeler may use this drop-down to select the correct type of dependency. The tool then automatically places a corresponding element within the dependency relation. Having defined the satisfaction of all goal elements in an iStar model, the modeler may execute an algorithm that traverses the goal hierarchy and computes an overall goal satisfaction value (Requirement [3]). Requirement [4] is addressed by an algorithm that runs with one or more models and generates a matrix for all intentional actors with their relationships, denoting depender, dependum, and the status of the relation (committed, open, critical). For requirement [5], the *view* mode of ADOxx has been utilized as the modeling languages used by the two iStar model types are heavily overlapping. The latter refines the former entirely or only in parts by using the same instance objects (Schwab et al., 2010).

5.2 The EnterKnow Case

The EnterKnow project[6] develops a Business Process Management System (BPMS) whose front-end behavior is determined, at run-time, by a back-end hybrid knowledge base semantically integrating several information sources: model contents, run-time data and open geographical data that is available through a variety of services (e.g., Google Maps[7]). All of these are integrated through Linked Data techniques (Heath and Bizer, 2011) in a common graph database, resulting in hybrid knowledge graphs further extended by reasoning mechanisms - OWL axioms and GeoSPARQL-based rules[8].

The modeling language at the core of this complex artifact is a domain-specific business process modeling language tailored to incorporate concepts from the Who, How and Where facets of the Zachman Framework (Zachman, 1987). Model contents capturing knowledge across these dimensions are exposed to the end-user's run-time software through the interoperability plug-in that exports model contents in the RDF format for linking and reasoning (e.g., OWL axioms, GeoSPARQL). The software engineering method devised and tested for this project - right-side of Figure 6, introduced and analyzed in (Buchmann et al., 2018) - is an instantiation of OMiLAB's "model-aware software development" process (see Figure 3), using RDF as an interoperability format and GraphDB's REST API[9] as an interoperability channel. The output product will be further evolved – e.g., to assimilate additional Zachman Framework facets with a granularity that is relevant to evolving requirements.

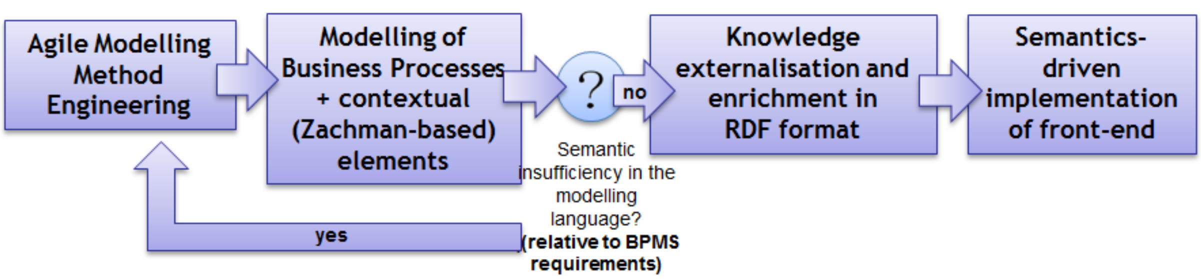

Figure 6. The EnterKnow model-aware software engineering process

The following specific requirements can be summarized for EnterKnow:

1. To extend a business process modeling language with several of the perspectives recommended by the Zachman Framework – more precisely their organizational context ("Who") and geographical locations ("Where");
2. To utilize openly available geographical data (and potentially any other Linked Open Data that can be connected to model elements in a meaningful way);
3. To minimize the inherent conceptual redundancy between the data model employed at run-time (by a BPMS's database) and the metamodel that governs the modeling tool at design-time;
4. To enable integration of model contents and run-time data in a way that is amenable to reasoning across the hybrid knowledge derived from this integration.

[6] EnterKnow Project page [online], http://austria.omilab.org/psm/content/enterknow/info?view=home, last checked: 2018-11-29
[7] Google Maps Platform [online], https://cloud.google.com/maps-platform/, last checked: 2018-11-27
[8] Open GeoSpatial Consortium, GeoSPARQL [online], http://www.opengeospatial.org/standards/geosparql,last checked: 2010-11-29
[9] Ontotext. 2018. GraphDB Downloads and Resources. [online], http://graphdb.ontotext.com, last checked: 2018-11-28

The EnterKnow prototype satisfies requirement [3] by assimilating, in the modeling language agilely tailored for requirement [1], parts of the data model that is employed by a BPMS at run-time. This partially fills the semantic gap that inherently exists between traditional model-driven software and the standard modeling languages that provide an invariant semantic space. The novel approach thus empowers modelers with the ability to apply diagrammatic configurations for the run-time behavior of software artifacts. The project reconsiders the traditional roundtrip engineering cycles through the lens of semantic technology and machine-readable knowledge flows.

Requirements [2] and [4] are satisfied with the help of Linked Data-based interoperability mechanisms that have been detailed in (Karagiannis & Buchmann, 2018):

- An ADOxx RDFizer plug-in and an RDF vocabulary for exporting any diagrammatic model. Transformation patterns to support generalization and potential adaptation for other metamodeling platforms have been published in (Karagiannis & Buchmann, 2016). The RDFizer becomes an adapter for semantic lifting, contributing to a growing list of such adapters that have been developed for legacy information sources, e.g. (Langegger & Wöß, 2009). It is thus a reusable resource made available in OMiLAB - earlier domain-specific applications have been reported for traceability of mobile app requirements and for enriching queries in the Internet of Things - see (Buchmann & Karagiannis, 2016; Buchmann & Karagiannis, 2017; Buchmann, 2016);
- Mechanisms for establishing semantic links between diagrammatic elements and external resources of various kinds: terms from existing vocabularies or ontologies, data entities from external Linked Open Data repositories, the possibility to attach, to any diagrammatic element, arbitrary RDF statements that are not restricted by the modeling language (i.e., by the metamodel);

These mechanisms enrich the Agile Modeling Method Engineering framework with additional steps to achieve a full-fledged software development method (see Buchmann et al., 2018) or, if approached from a Knowledge Management perspective, a knowledge conversion cycle that supports the development of knowledge-driven digital products (Karagiannis et al., 2017).

6 Related Approaches

Method engineering has a long tradition in information science, computer science and systems development (cf. Brinkkemper, 1996; Nuseibeh et al., 1996; Ralyté et al., 2003; Rolland, 2009). In recent years, it has also been considered as an important area of information science research (Recker, 2015), focusing on the development of domain-specific conceptual modeling methods (Frank, 2013; Frank et al., 2014). Due to its ambiguous usage (Henderson-Sellers & Ralyté, 2010), method engineering needs to be tailored to the notion of conceptual modeling and model value employed by the OMiLAB (cf. Figures 2 and 3).

In contrast to the more general term "method engineering" used in systems engineering, where situation-specific methods are established by combining existing method chunks, in conceptual modeling method engineering is understood as the process of creating a new *modeling method* with the aim of creating conceptual models as abstract descriptions of some real world phenomenon to foster understanding and support communication (which we extend to "interoperability") as proposed by Mylopoulos (1992). This creation of modeling methods is realized in the OMiLAB by the conceptualization process (see Figure 3 at the top) whose building blocks target the design of a new conceptual modeling method, aligned to domain- or case-specific requirements. Moreover, OMiLAB facilitates the operationalization of outcomes with the help of specialized kinds of resources and services made available in its digital ecosystem (see Figure 3 at the bottom).

Besides the primary scope of the OMiLAB as described above, approaches aiming to establish open communities in the fields of method engineering and conceptual modeling are considered related. For example, (Koch et al., 2006) aimed for building a community that produces and shares reference models for software development. Several other community-oriented initiatives have been established in recent years. Table 3 provides a brief description of the most relevant ones and contrasts them to the OMiLAB.

Table 3. Overview of related approaches on method engineering and community building in IS

Approach	Description
DSM Forum	"The DSM Forum exists to spread the knowledge and know-how of Domain-Specific Modeling. It is an independent body made up of the leading DSM tool and solution providers, along with expert DSM users". The DSM Forum homepage lists commercial and open DSM tools, DSM publications, and DSM case studies. The forum is not restricted to a certain application domain like conceptual modeling or model-driven engineering. Moreover, it is not limited to a certain modeling tool or development platform. The focus of the DSM Forum is on providing content specifically of benefit for researchers and practitioners. Consequently, it does not provide any learning materials or trainings. Moreover, there is no active community behind the DSM Forum, it is merely a repository of relevant information and technologies useful for modeling method conceptualization.
Free Models Initiative	The Free Models Initiative was established in 2014 by a kick-off Workshop organized at the 'Modellierung' conference. The vision of the initiative is to provide free model corpora for researchers in the domains of software engineering, model-driven development, and business process management. The corpora shall enable scientific benchmarks, evaluations and research model-based approaches. The initiative does not provide modeling tools, nor does it address educational aspects. The initiative homepage lists existing model corpora. This initial activity should lead to a living community. Despite the interesting and relevant goal, the initiative seems to be already inactive at the time of this paper's writing. It can be asserted that this initiative aims at providing open models for an open community. However, in contrast to the OMiLAB, this initiative is neither interested in defining new modeling methods, nor the realization and provision of modeling tools. Lastly, the initiative is focusing on providing model corpora for research purposes.
GenMyModel	In January 2013, the GenMyModel initiative started to establish a world-wide community of software engineers (Dirix et al., 2013). The platform provides basic functionality to collaboratively create models relevant for software engineering. The platform currently supports the following modeling languages: ArchiMate, BPMN, DMN; UML, RDS, and Flowchart. The strength of GenMyModel is the low entry barrier for modelers by its browser-based online modeling and collaboration environment. The platform comes with a free sign-up package (limited model size, limited collaborators etc.), but for reasonable usage, a monetary subscription is mandatory. The created models can be shared and discussed in the community. In contrast to the OMiLAB, this approach fosters on the modeling part of de-facto industry standards built on an economic business model. An overview of similar modeling-as-a-service approaches for software engineering can be found in (Di Rocco et al., 2015; Popoola et al., 2017).
WeST	The Institute for Web Science and Technologies was founded in 2009. It aims to establish a multi-perspective research framework on topics related to the digital transformation. A clear focus thereby is on web services and web technologies like semantic web, data mining, interactive web, software and services in the web. Additional working groups focus on digital markets, e-government, and digital entrepreneurship. The spectrum of covered topics is impressive. Besides, WeST also incorporates a huge number of international industry and research partners. Research is condensed in research laboratories, located mostly in Europe with facilities also in North and South America, and Asia. WeST acquires a lot of funding budget for research projects, mostly in the German-speaking and European area. WeST also focuses on education. Since 2016, an annual WSTNet Web Science Summer School is organized. Moreover, the University of Koblenz-Landau offers a Master's program in Web Science in cooperation with the WeST. WeST has a strong focus on applied research that provides a benefit for the industry. It does not employ or share a set of technologies. Expert knowledge in web technologies and semantic technologies is mandatory to contribute to the community. It is not clear how interested researchers can collaborate.

Based on the presented related approaches can be summarized, that to the best of our knowledge, there is no initiative comparable to the OMiLAB by means of having an identical scope, sharing a similar vision of openness (Bork and Miron, 2017) and a rich toolset of enablers. OMiLAB focusses on a co-creation strategy and its required digital ecosystem related to conceptual modeling methods. All presented initiatives have their strengths in isolated aspects but do not provide a holistic view particularly on the conceptualization and operationalization of conceptual modeling methods. We see this research also as a means of starting to collaborate with those initiatives in order to bundle the strengths and to bridge the information and computer science modeling communities.

7 Evaluation

The OMiLAB comprises a multitude of tools, services, best practices as well as a repository of reusable tooling and knowledge assets pertaining to the conceptualization and operationalization of modeling methods. The case studies presented previously showed feasibility of the OMiLAB for modeling method conceptualization. In the following, two additional evaluations are reported. First, quantitative measures will be given on i) the practical impact of the OMiLAB by referring to OMiLAB modeling tool downloads, and ii) the educational impact of the OMiLAB by referring to the Next-generation Enterprise Modeling Summer School. Second, the learned lessons from a failed attempt to establish an open ecosystem (Briel & Recker, 2017) will be used to evaluate the extent to which the OMiLAB is prepared for them.

7.1 Quantitative Evaluation

7.1.1 Practical Impact

Using the OMiLAB digital ecosystem, almost 50 modeling methods have been realized by research groups distributed around the world (Karagiannis et al., 2016). All methods and the corresponding modeling tools are hosted on the OMiLAB Portal (see Section 4.3). The methods are tailored for a variety of project-based use cases and domain-specific requirements, both educational and research-oriented.

Practical impact of the OMiLAB is evaluated in the following by looking at the download numbers for the realized open modeling tools. In July 2016, the new OMiLAB portal was deployed and is since then in operation. As part of this portal, a File Manager microservice handles all file downloads including the modeling tool downloads. In this period of 29 months, 8.277 downloads have been recorded (see Figure 7). Consequently, every month approximately 285 OMiLAB tools were downloaded and used worldwide in education or research. The peaks in March indicate, that the tools are frequently also used in university courses, as most European universities start their summer term semester in March/April. The peak in July 2018 indicates the interest of the NEMO 2018 students to download the tools corresponding to the lectures. As of November 2018, 18 modeling tools were already downloaded more than 100 times, 31 tools more than 30 times.

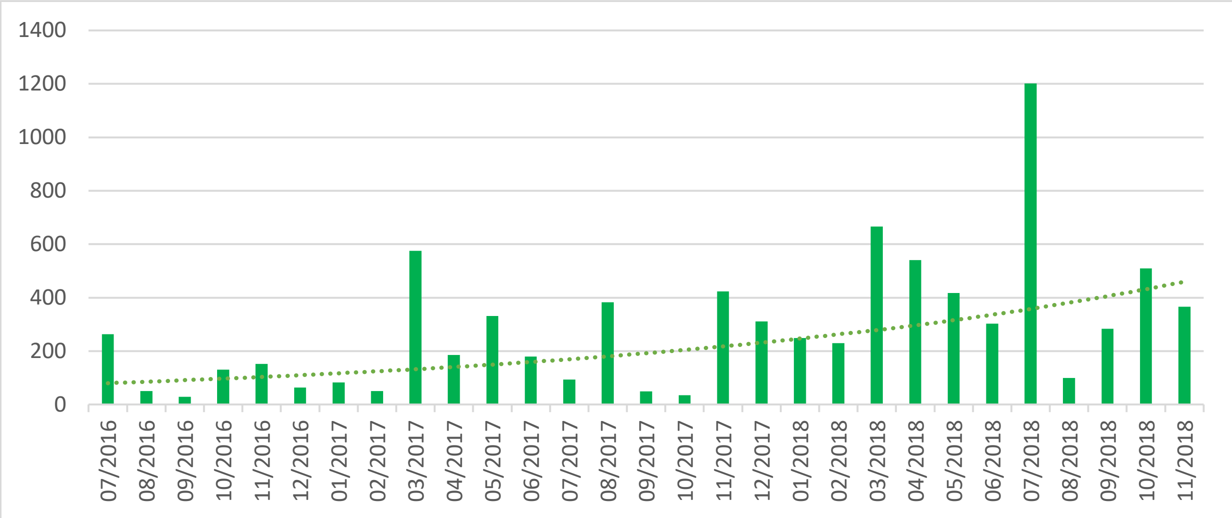

Figure 7. OMiLAB tool downloads from July 2016 until November 2018

It should be noted, that these numbers establish the lower bound of the actual download and usage numbers. Community members are of course allowed to also upload their tools and provide them to their students via local learning management systems like Moodle or private webpages. Thus, it is very likely that the actual practical impact is even higher. The OMiLAB encourages collective intelligence for the common good (cf. Schuler et al., 2015). As the realized tools gain maturity and the methods publicity, the OMiLAB network establishes a knowledge base for modeling methods, open tools, and freely available teaching materials.

7.1.2 Educational Impact

Education-wise, OMiLAB resources cover multiple abstraction layers and application domains for teaching conceptual modeling as a self-contained disciplined and in a consistent manner. By subordinating modeling practices to other disciplines, the general value of conceptual models tends to be fragmented and reduced to isolated (albeit popular) use cases. OMiLAB stimulates a top-down holistic view which may be further specialized across study programs for their relevant application domains. As a quantifiable measure for the educational impact of the OMiLAB, we use the NEMO Summer School series in the following. In its first five years, NEMO featured 197 theoretical lectures and numerous practical exercises. In total, 284 students have been educated in manifold aspects of conceptual modeling (see Table 4).

Table 4. Students and Lecturers of the NEMO Summer School series

		2014	2015	2016	2017	2018
Students	Number of	50	55	67	57	55
	Institutions	29	26	43	39	38
	Countries	15	16	28	25	25
Lecturers	Number of	33	38	43	45	38
	Institutions	26	32	35	33	31
	Countries	15	20	19	18	19

A detailed qualitative survey among 75 students of NEMO 2016 (cf. Bork et al., 2016) and 2018 shows that students appreciate teaching conceptual modeling in interesting domains like Smart Cities, which is the base for one of the core exercises in recent NEMO editions and also part of the OMiLAB training material on metamodeling. Students appreciate the usage of open tools in the practical sessions as a means of gaining hands-on modeling experience. Considering the overall evaluation of the NEMO Summer School, an average value of 4.6667 (the highest response in the survey) reflects that 72 of the 75 students mostly agree or fully agree that they would recommend participating in future NEMO editions to their colleagues or other students.

7.2 Literature-based Learned Lessons

We relate to the study of (Briel & Recker, 2017) which, starting from identified pitfalls and fallacies has synthesized a set of learned lessons for online open innovation communities. In the following we ground those learned lessons in the case of OMiLAB:

Lesson 1. Acknowledge the Industry Context. This is currently only a marginal concern since the targeted stakeholders are primarily academic researchers and lecturers who embrace the philosophy of open use, open access and open source, being interested in spreading their ideas and in widening the visibility of their projects. Industry stakeholders are typically indirectly involved as project partners – however, intellectual property for OMiLAB contributions is retained by their respective source-projects and contributors, each of them deciding which parts to contribute according to their project-level agreements. In NEMO 2018, we actually welcomed - for the first time - participants coming from industry.

OMiLAB is managed by a non-profit organization and the fostered innovation is aimed to be a public good, to facilitate the state-of-the-art enrichment and to push a teaching agenda for conceptual modeling as a mature discipline that produces specific kinds of value. As industry stakeholders may show more active involvement, it is plausible that requirements could be raised to partition the community in different sections. The in-house content management system underlying the OMiLAB portal is prepared in terms of modularity and extensibility for such an evolution.

Lesson 2. Consider the Legal and Regulatory Environment. OMiLAB aims to grow as an international network of OMiLAB installations (e.g. OMiLAB Korea) that provide geographical coverage and adaptation to regional regulatory environments. Currently, data is stored in Europe and policies for complying to GDPR regulations are being refined. A non-profit organization was established to manage the infrastructure and administrative aspects pertaining to OMiLAB events.

Lesson 3. Establish Support Processes Early. Support is being developed on a continuous basis in the form of services that facilitate various conceptualization steps (e.g., design of graphics, model annotation, plug-ins, IDEs), as well as tutorials and demos of varying specificity for training on each conceptualization step of the AMME framework. The OMiLAB portal is built on a microservice Content-Management System architecture that allows new project spaces to be quickly set-up, with adequate granularity and extensible sections. Dedicated personnel is maintaining this to ensure responsiveness for the needs of community members.

Also, the AMME framework and the modeling method "seed concept" have been designed in a modular way to avoid heavyweight and rigid implementation processes, allowing contributors to become involved in the steps where they hold most interest (e.g., picking an open method specification and deploying it, replacing the notation of a modeling language, integrating web services and third-party tools).

Lessons 4-5. Prepare to Shift the Organizational Mindset & Get Ready to Adapt Your Current Innovation Approach. It is still too early to assess the evolution of this digital ecosystem, but it should be in line with how conceptual modeling evolves as a research paradigm, with milestones such as the study of (Sabegh & Recker, 2017b). Requirements are gradually derived from such studies and will motivate the future strategies of OMiLAB.

Lesson 6. Know Your Contributors. OMiLAB stakeholders meet frequently either in dedicated OMiLAB events (the NEMO Summer School series, OMiLAB-focused workshops attached to various conferences) or in community-specific scientific conferences (typically those that touch on conceptual modeling topics: ER, PoEM, CAISE, MODELS etc.). Project-based research visits are being hosted on a regular basis by the physical space of OMiLAB at University of Vienna, and also facilitated at the site of various contributors. Each project that has been hosted on the OMiLAB portal included such experience exchanges where stakes are being clarified in order to refine the scope and requirements for OMiLAB.

8 Conclusion

The Open Models Laboratory (OMiLAB, http://omilab.org/) is a digital ecosystem for the conceptualization and operationalization of modeling methods, combining open source and open community principles with the goal of fostering cross-domain practice-oriented information science and design science research. A repository of all realized modeling tools is available online[10]. The methods and tools provided by the OMiLAB enable research, application, and education in conceptual modeling. Moreover, OMiLAB allows researchers to deploy their method artifacts in order to have proofs-of-concepts published and evaluated.

This paper introduced the fundamental pillars of the OMiLAB, i.e.; (a) a granularly defined "Modeling Method" concept whose building blocks can be customized for the domain of choice; (b) an "Agile Modeling Method Engineering" framework supporting quick prototyping of modeling tools; (c) a model-aware “Digital Product Design Lab”; and (d) dissemination channels for reaching a global community. Based on two application cases, viability of the OMiLAB was demonstrated. Moreover, a comprehensive quantitative evaluation has been performed to highlight the practical and the educational impact of the OMiLAB. OMiLAB acts as a successful best practice in how to establish an online innovation community. It is important to note, that the conceptual contributions presented in this paper, i.e., the OMiLAB Digital Ecosystem with its conceptualization (Figure 3 top) and operationalization (Figure 5) of modeling methods already have physical installments (as shown in the Figure 8).

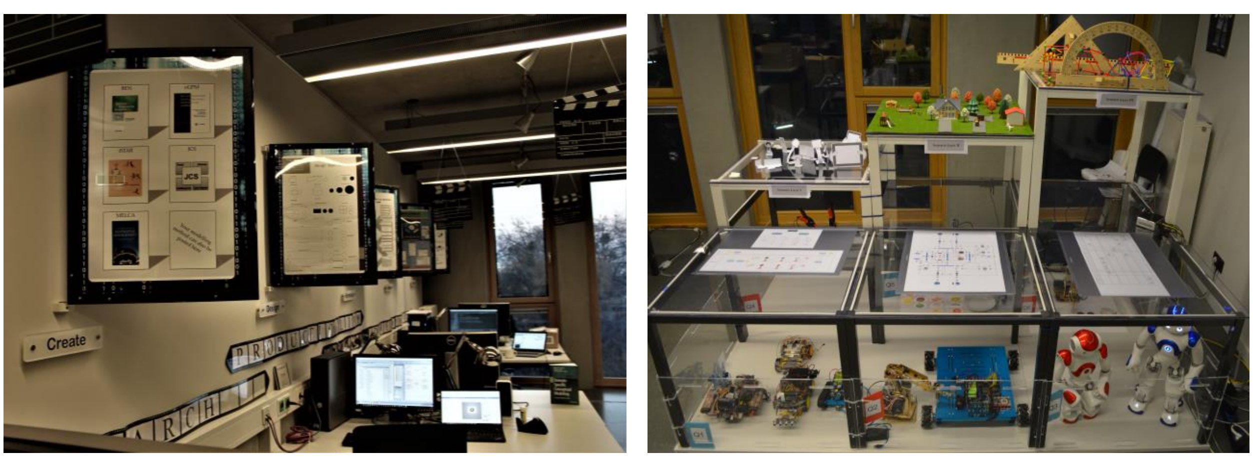

Figure 8. Physical installements of the Conceptualization (left) and the Digital Product Space (right)

[10] Modeling tool repository [online] http://austria.omilab.org/psm/exploreprojects?param=explore last checked:2018-11-29

Generally, three OMiLAB usage scenarios can be differentiated:

- One is able to find a modeling method in OMiLAB that suits all requirements. In this case, one can simply **download and use** the tool.
- An existing method in OMiLAB addresses some of the requirements. In this case, the method can be **extended** to fulfill all requirements (cf. Sabegh and Recker (2017a)).
- No method fits to the requirements. In this case, the OMiLAB can guide the **conceptualization** of a novel modeling method specifically designed to address the requirements.

By motivating more researchers worldwide to contribute their platforms, methods and tools, we are confident to diversify the technological landscape of the ecosystem. Currently, OMiLAB community members had little interest in migrations of method specifications across different metamodeling platforms and challenges derived from such efforts have yet to be documented. A recent research project started to analyze ADOxx-based metamodels in comparison with Ecore-based ones (Bork, 2018) with the goal of supporting technological openness and migration.

Referring back to our research question, this paper showed results of our practical work in supporting modeling method engineers. Based on the 50 realized OMiLAB projects, we are confident, that the digital ecosystem provides the necessary tools and services to foster modeling method engineering. However, we need to empirically investigate whether the procedure applied and the tools/services used can be further improved. We are planning to involve the OMiLAB community in this evaluation process more directly.

Future work will concentrate on further extending the OMiLAB by: i) involving new community members with their domains (e.g., robotics, Industry 4.0); ii) attracting developers of further tools and services supporting modeling method conceptualization in order to improve project success (cf. Jetu & Riedl, 2012); iii) inauguration of further OMiLAB nodes worldwide to enhance the global network whilst enabling local bonding; and iv) to advocate the relevance of conceptual modeling in the IS curriculum by establishing NEMO as a key educational forum that propagates ideas and contents towards national-level study programs.

ACKNOWLEDGMENTS

The work of Robert Buchmann was supported by the Romanian National Research Authority through UEFISCDI, under grant agreement PN-III-P2-2.1-PED-2016-1140.

References

Becker, J., vom Brocke, J. Heddier, M. & Seidel, S. (2015). In Search of Information Systems (Grand) Challenges. *Business & Information Systems Engineering*, 57(6), 377-390.

Benlian, A., Hilkert, D. & Hess, T. (2015). How open is this platform? The meaning and measurement of platform openness from the complementors perspective. *Journal of Information Technology*, 30(3), 209-228.

ADOxx.org (2018). ADOxx Official Page [online], http://adoxx.org, last checked: 2018-12-06

Bock, A. & Frank, U. (2016). *Multi-perspective Enterprise ModelingConceptual Foundation and Implementation with ADOxx*. In Karagiannis, D., Mayr, H. C & Mylopoulos, J. (Eds.) *Domain-Specific Conceptual Modeling* (pp. 241-267). Springer.

Bork, D. & Sinz, E. J. (2010). Design of a SOM Business Process Modelling Tool based on the ADOxx meta-modelling Platform, In de Lara, J. & Varro, D. (Eds.), *Pre-Proceedings 4th international Workshop on Graph-Based Tools (GraBaTs)* (pp. 89-101). NL-Enschede.

Bork, D., & Fill, H. G. (2014). Formal aspects of enterprise modeling methods: a comparison framework. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on* (pp. 3400-3409). IEEE.

Bork, D., Fill, H. G., Karagiannis, D., Miron, E. T., Tantouris, N., & Walch, M. (2015). Conceptual Modelling for Smart Cities: A Teaching Case. *Interaction Design & Architectures (IxD&A)*, Special Issue on Smart City Learning: Opportunities and Challenges, 27, 10-28.

Bork, D., Buchmann, R., Hawryszkiewycz, I., Karagiannis, D., Tantouris, N., & Walch, M. (2016). Using Conceptual Modeling to Support Innovation Challenges in Smart Cities. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on* (pp. 1317-1324). IEEE.

Bork, D., Karagiannis, D. & Hawryszkiewycz, I. (2017). Supporting Customized Design Thinking Using a Metamodel-based Approach. In *28th Australasian Conference on Information Systems (ACIS 2017)*, Hobart, Australia.

Bork, D., & Miron, E. T. (2017). OMiLAB - An open innovation community for modeling method engineering. In *8th International Conference of Management and Industrial Engineering* (pp. 64-77).

Bork, D. (2018). Metamodel-based Analysis of Domain-specific Conceptual Modeling Methods. 11th IFIP WG 8.1 working conference on the Practice of Enterprise Modelling (PoEM), LNBIP 335, In press.

Bork, D., Karagiannis, D., & Pittl, B. (2018). Systematic analysis and evaluation of visual conceptual modeling language notations. In *2018 12th International Conference on Research Challenges in Information Science (RCIS)* (pp. 1-11). IEEE.

Bork, D., Karagiannis, D. & Pittl, B. (2018b): How are Metamodels Specified in Practice? Empirical Insights and Recommendations, In *24th Americas Conference on Information Systems (AMCIS'2018)*, 16-18 August 2018, New Orleans, USA, in press.

Breitling, H. & Hofer, S. (2012). Beispielhaft gut modelliert: Exemplarische Geschäftsprozess-Modellierung in der Praxis. *OBJEKTspektrum*, 6, 8-13.

Brenner, W., Karagiannis, D., Kolbe, L., Krüger, J., Leifer, L., Lamberti, H-J., Leimeister, J. M., Österle, H., Petrie, C. & Plattner, H. (2014). User, use & utility research. *Business & Information Systems Engineering*, 6(1), 55-61.

Briel, F. von & Recker, J. (2017). Lessons from a failed implementation of an online open innovation community in an innovative organization. *MIS Quarterly Executive* 16(1), 35-46.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and software technology*, 38(4), 275-280.

Bucher, T., Klesse, M., Kurpjuweit, S. & Winter, R. (2007). *Situational Method Engineering*. In Situational method engineering: fundamentals and experiences (pp. 33-48). Springer.

Buchmann, R. A. (2016). Modeling Product-Service Systems for the Internet of Things: The ComVantage Method. In Karagiannis, D., Mayr, H. C & Mylopoulos, J. (Eds.) *Domain-Specific Conceptual Modeling* (pp. 417-437). Springer.

Buchmann, R. A. & Karagiannis, D. (2015). Agile Modelling Method Engineering: Lessons Learned in the ComVantage Research Project. In Ralyte, J., Espana, S., Pastor, O. (eds) *Proceedings of the 8th IFIP WG 8.1 Working Conf. on the Practice of Enterprise Modeling (PoEM 2015),* LNBIP 235, Springer, pp. 356-373.

Buchmann, R. A. & Karagiannis, D. (2016). Enriching Linked Data with Semantics from Domain-Specific Diagrammatic Models. *Business & Information Systems Engineering*, 58(5), 341-353.

Buchmann, R. A. & Karagiannis, D. (2017). Modelling mobile app requirements for semantic traceability. *Requirements Engineering*, 22(1), 41-75.

Buchmann, R. A. & Ghiran, A. M. (2017). Engineering the Cooking Recipe Modelling Method: a Teaching Experience Report, In: *Proceedings of PrOse 2017*, CEUR-WS vol. 1999, paper 5.

Buchmann, R.A., Cinpoeru, M., Harkai, A. & Karagiannis, D. (2018). Model-Aware software engineering: a knowledge-based approach to Model-Driven software engineering, In Damiani, E., Spanoudakis, G. & Maciaszek, L (eds.) *Proceedings of the 13th Int. Conf. on Evaluation of Novel Approaches to Software Engineering (ENASE)*, SCITE Press, 233-240

Cairó, O. & Guardati, S. (2012). The KAMET II methodology: Knowledge acquisition, knowledge modeling and knowledge generation. *Expert Systems with Applications*, 39(9), 8108-8114.

Cairó Battistutti, O. & Bork, D. (2017): Tacit to Explicit Knowledge Conversion, *Cognitive Processing Journal*, in press.

Delfmann, P., Breuker, D., Matzner, M. & Becker, J. (2015). Supporting Information Systems Analysis Through Conceptual Model Query – The Diagramed Model Query Language (DMQL). *Communications of the Association for Information Systems*, 37, Article 24.

Dirix, M., Muller, A., & Aranega, V. (2013). Genmymodel: an online uml case tool. In European Conference on Object-Oriented Programming (ECOOP).

Di Rocco, J., Di Ruscio, D., Iovino, L., & Pierantonio, A. (2015). Collaborative repositories in model-driven engineering [software technology]. *IEEE Software*, (3), 28-34.

Efendioglu, N., Woitsch, R., & Utz, W. (2016). A Toolbox Supporting Agile Modelling Method Engineering: ADOxx. org Modelling Method Conceptualization Environment. In *IFIP Working Conference on The Practice of Enterprise Modeling* (pp. 317-325). Springer, Cham.

Ferstl, O. K., Sinz, E. J. & Bork, D. (2016). Tool Support for the Semantic Object Model. In Karagiannis, D., Mayr, H. C & Mylopoulos, J. (Eds.) *Domain-Specific Conceptual Modeling* (pp. 291-310). Springer.

Fill, H-G., Redmond, T. & Karagiannis, D. (2012). FDMM: A formalism for describing ADOxx meta models and models. In Maciaszek, L., Cuzzocrea, A. & Cordeiro, J. (Eds.) *Proceedings of ICEIS 2012 - 14th International Conference on Enterprise Information Systems, Vol.3*, 133-144.

Franch, X., López, L., Cares, C. & Colomer, D. (2016). The i* Framework for Goal-Oriented Modeling. In Karagiannis, D., Mayr, H. C & Mylopoulos, J. (Eds.) *Domain-Specific Conceptual Modeling* (pp. 485-506). Springer.

Frank, U. (2013). Domain-specific modeling languages: requirements analysis and design guidelines. In *Domain Engineering* (pp. 133-157). Springer.

Frank, U., Strecker, S., Fettke, P., vom Brocke, J., Becker, J. & Sinz, E. J. (2014). The research field modeling business information systems. *Business & Information Systems Engineering*, 6(1), 39-43.

Galvagno, M. & Dalli, D. (2014). Theory of value co-creation: a systematic literature review, *Managing Service Quality*, Vol. 24 Issue: 6, pp.643-683

Hawryszkiewycz, II. T. & Prackwieser, C. (2016). MELCA - Customizing Visualizations for Design Thinking. In Karagiannis, D., Mayr, H. C & Mylopoulos, J. (Eds.) *Domain-Specific Conceptual Modeling* (pp. 383-396). Springer.

Heath, T. & Bizer, C. (2011) Linked Data: Evolving the Web into a Global Data Space (1st edition). Morgan & Claypool, San Rafael, USA

Henderson-Sellers, B. & Ralyté, J. (2010). Situational method engineering: State-of-the-art review. *Journal of Universal Computer Science*, 16(3), 424-478.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75-105.

Höfferer, P. (2007). Achieving business process model interoperability using metamodels and ontologies. In *Proceedings of the 15th European Conference on Information Systems (ECIS 2007)* (pp. 1620-1631).

Horkoff, J. & Yu, E. (2016). Interactive goal model analysis for early requirements engineering. *Requirements Engineering*, 21(1), 29-61.

Jarke, M., Gallersdörfer, R., Jeusfeld, M. A., Staudt, M. & Eherer, S. (1995). ConceptBase - a deductive object base for meta data management. *Journal of Intelligent Information Systems*, 4(2), 167-192.

Jetu, F. T. & Riedl, R. (2012). Determinants of Information Systems and Information Technology Project Team Success: A Literature Review and a Conceptual Model. *Communications of the Association for Information Systems*, 30, Article 27.

Jeusfeld, M. A. (2016). SemCheck: Checking Constraints for Multiperspective Modeling Languages. In Karagiannis, D., Mayr, H. C & Mylopoulos, J. (Eds.) *Domain-Specific Conceptual Modeling* (pp. 31-53). Springer.

Karagiannis, D. (2015). Agile modeling method engineering. In *Proceedings of the 19th Panhellenic Conference on Informatics* (pp. 5-10). ACM.

Karagiannis, D. & Buchmann, R. A. (2016). Linked Open Models: Extending Linked Open Data with conceptual model information. *Information Systems*, 56, 174-197.

Karagiannis, D., Buchmann, R. A., & Bork, D. (2016a). Managing Consistency in Multi-View Enterprise Models: an Approach based on Semantic Queries. In *Proceedings of ECIS 2016* (p. ResearchPaper 53), AIS.

Karagiannis, D. Buchmann, R. A., Burzynski, P., Reimer, U. & Walch, M. (2016b). Fundamental Conceptual Modeling Languages in OMiLAB. In *Domain-Specific Conceptual Modeling* (pp. 3-30).

Karagiannis, D., Grossmann, W. & Höfferer, P. (2008). *Open Model Initiative: A Feasibility Study*.

Karagiannis, D. & Kühn, H. (2002). Metamodeling Platforms. In Bauknecht, K., Min Tjoa, A. & Quirchmayr, G. (Eds.) *Third International Conference EC-Web 2002 Dexa 2002* (pp. 182), Aix-en-Provence, France. Springer.

Karagiannis, D., Mayr, H. C. & Mylopoulos, J. (2016). *Domain-Specific Conceptual Modelling*. Springer.

Karagiannis, D., Buchmann, R. A., Walch, M. (2017). How can Diagrammatic Conceptual Modelling Supporting Knowledge Management? In Proceedings of ECIS 2017 (Paper 101), AIS.

Karagiannis, D. & Buchmann, R.A. (2018) A proposal for deploying hybrid knowledge bases: the ADOxx-to-GraphDB interoperability case. In *Proceedings of the 51st Hawaii Conference on System Sciences* (HICSS), p. 4055-4064, University of Hawaii.

Koch, S., Strecker, S. & Frank, U. (2006). Conceptual modelling as a new entry in the bazaar: The open model approach. In *IFIP International Conference on Open Source Systems* (pp. 9-20). Springer.

Langegger, A. & Wöß, W. (2009). XLWrap - querying and integrating arbitrary spreadsheets with SPARQL. In *International Semantic Web Conference* (pp. 359-374). Springer.

Li, T., Grubb, A. M. & Horkoff, J. (2016). Understanding Challenges and Tradeoffs in iStar Tool Development. In López, L. & Yu, Y (Eds.) *Proceedings of the Ninth International i* Workshop co-located with 24th International Conference on Requirements Engineering (RE 2016)* (pp. 49-54), Beijing, China. CEUR-WS.org.

Miron, E.-T., Muck, C., Karagiannis, D., & Götzinger, D. (2018). Transforming Storyboards into Diagrammatic Models. In *International Conference on Theory and Application of Diagrams*, (pp. 770-773) Springer, Cham.

Miron, E.-T., Muck, C., & Karagiannis (2019). Transforming Haptic Storyboards into Diagrammatic Models: The Scene2Model Tool. In, *52th Hawaii International Conference on System Sciences* (in press), Grand Wailea, Maui.

Moody, D. (2009). The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6), 756-779.

Mylopoulos, J. (1992). 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), New York: John Wiley & Sons.

Nuseibeh, B., Finkelstein, A. & Kramer, J. (1996). Method engineering for multi-perspective software development. *Information and Software Technology*, 38(4), 267-274.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A. & Sinz, E.J. (2011). Memorandum on design-oriented information systems research, In *European Journal of Information Systems* 20(1):7-10.

Popoola, S., Carver, J., & Gray, J. (2017). Modeling as a Service: A Survey of Existing Tools. In *MODELS (Satellite Events)* (pp. 360-367).

Ralyté, J., Deneckère, R. & Rolland, C. (2003). Towards a Generic Model for Situational Method Engineering. In Eder, J. & Missiko, M. (Eds.) *Advanced Information Systems Engineering: 15th International Conference* (pp. 95-110), Klagenfurt/Velden, Austria, Springer Berlin Heidelberg.

Recker, J. (2015). Research on conceptual modelling: less known knowns and more unknown unknowns, please. In Saeki, M. & Koehler, H. (Eds.) *Proceedings of the 11th Asia-Pacific Conference on Conceptual Modelling*, Australian Computer Society, Sydney, Australia (pp. 3-7).

Rolland, C. (2009). Method engineering: towards methods as services. *Software Process: Improvement and Practice*, 14(3), 143-164.

Sabegh, M. A. J. & Recker, J. (2017a). Combined Use of Conceptual Models in Practice: An Exploratory Study. *Journal of Database Management (JDM)*, 28(2), 56-88.

Sabegh, M. A. J., Lukyanenko, R., Recker, J., Samuel, B., & Castellanos, A. (2017b). Conceptual modeling research in information systems: What we now know and what we still do not know. In 16th AIS SIGSAND Symposium, 19-20 May 2017, Cincinnati, OH.

Sandkuhl, K., Fill, H. G., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A., Schwabe, G., Uludag, Ö, Winter, R. (2018). From expert discipline to common practice: a vision and research agenda for extending the reach of enterprise modeling. *Business & Information Systems Engineering* 60(1):69-80.

Schuler, D., De Cindio, F. & De Liddo, A. (2015). Encouraging Collective Intelligence for the Common Good: How Do We Integrate the Disparate Pieces? In *Proceedings of the 7th International Conference on Communities and Technologies, ComTech '15* (pp. 157-159), New York, USA. ACM.

Schwab, M., Karagiannis, D. & Bergmayr, A. (2010). i* on ADOxx: A Case Study. In Castro, J., Franch, X., Mylopoulos, J. & Yu, E. (Eds.) *Proceedings of the 4th International i* Workshop - iStar10* (pp. 92-97). Citeseer.

Stark, J., Braun, R. & Esswein, W. (2017). Systemizing Colour for Conceptual Modeling. In Leimeister, J. M. & Brenner, W. (Eds.) *Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017)* (pp. 256-270), St. Gallen.

Utz, W., & Falcioni, D. (2018). Data Assets for Decision Support in Multi-Stage Production Systems Industrial Business Process Management using ADOxx. In *2018 IEEE 16th International Conference on Industrial Informatics* (pp. 809-814). IEEE.

van der Aalst, W. M. P., Bichler, M. & Heinzl, A. (2016). Open Research in Business and Information Systems Engineering. *Business & Information Systems Engineering*, 58(6), 375-379.

Visic, N., Fill, H-G., Buchmann, R. A. & Karagiannis, D. (2015). A domain-specific language for modeling method definition: From requirements to grammar. In *9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015* (pp. 286-297), Athens, Greece, IEEE.

Walch, M. & Karagiannis, D. (2017). Service-driven enrichment for KbR in the OMiLAB environment. In Hara, Y. & Karagiannis, D. (eds.) *Serviceology for Services (Proceedings of ICServ 2017)* (pp. 164-177), LNCS 10371, Springer.

Walch, M. & Karagiannis, D. (2019). How to connect design thinking and cyber-physical systems: the s*IoT conceptual modelling approach. Accepted for *System Sciences (HICSS), 2019 52nd Hawaii International Conference on*, IEEE.

Winter, R. (2010). *Business engineering navigator: Gestaltung und Analyse von Geschäftslösungen" Business-to-IT"*. Springer-Verlag.

Yu, E. S-K. (1995). *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto.

Zachman, J.A. (1987). A framework for information systems architecture, *IBM Systems Journal* 26 (3), 276-292.