Satisfying Four Requirements for More Flexible Modeling Methods: Theory and Test Case

Dominik Bork[*,a], Steven Alter[b]

[a] University of Vienna, Faculty of Computer Science, Research Group Knowledge Engineering, Vienna, Austria
[b] School of Management, University of San Francisco, San Francisco, USA

Abstract. *Recent research in conceptual modeling and enterprise modeling calls for relaxing common assumptions about the nature of modeling methods and related modeling languages and metamodels. This paper pursues that goal by proposing a new vision of modeling methods that overcomes some of the limitations identified in the literature by satisfying four requirements for more flexible modeling methods. That vision builds upon the integration of multiple modeling techniques that are related to an overarching metaphor. Those techniques may address heterogeneous purposes such as specifying a system's capabilities or specifying which resources are used by specific activities. This paper presents design characteristics and metamodel design options to guide method engineers in adopting this broader notion of modeling methods, integrating multiple modeling techniques, and using appropriate modeling languages. To demonstrate feasibility, an extended version of the work system method (WSM) is presented in the form of a Work System Modeling Method (WSMM) that encompasses seven purposes of modeling that call for successively more formal approaches. A final section summarizes how WSMM addresses the issues and requirements from the introduction, explains how coherence is maintained within WSMM, and identifies areas for future research, with emphasis on ways to make WSMM and similar modeling methods as valuable as possible.*

Keywords. enterprise modeling • metamodeling • work system method • work system theory • work system modeling method

Communicated by Peter Fettke. Received 2019-08-07. Accepted on 2020-01-17.

1 Introduction

The benefits of enterprise models often come at the cost of complexity and inflexibility due to formalization and rigor needs of modeling methods and supporting tools. In contrast, domain experts often perceive the business in imprecise ways and may or may not have the expertise to capture their knowledge in a conceptual model (cf. Bjeković et al. (2014), Figl (2017), Gonzalez-Perez (2018), and Zur Muehlen and Recker (2013)). Furthermore, modeling tools sometimes constrain intuitive specification of externalized knowledge by forcing users to express themselves in modeling languages that are unfamiliar or difficult to use (cf. Correia and Aguiar (2013) and Wüest et al. (2019)).

Prominent researchers from various backgrounds argue that modeling methods for enterprise and process modeling have not achieved their full potential and need to be extended or augmented to make them more usable by broader user groups. This paper's approach to modeling methods addresses important issues that they mention:

* Corresponding author.
E-mail. dominik.bork@tuwien.ac.at
Note: This work extends previous publications that introduced foundational ideas on the Work System Modeling Method (Bork and Alter (2018), Alter and Bork (2019)). It was processed as a Fast Track submission extending Bork and Alter (2019) which as published in the Proceedings of WI 2019.

- A Dagstuhl seminar (Clark et al. 2016) emphasized how differing stakeholder needs call for different approaches to enterprise modeling (EM). That seminar led to a *BISE* research note by EM community leaders (Sandkuhl et al. 2018) that encourages moving EM from an expert discipline towards “grass roots modeling” and “modeling for the masses.” Their future research agenda includes “softened requirements to completeness, coherence and rigor”.
- Six of seven process modeling problems discussed in Aalst (2012) are relevant here: 1) aiming for one model that suits all purposes, 2) straightjacketing smaller interactive processes into one monolithic model, 3) using static hierarchical decomposition as the only abstraction mechanism, 4) modeling humans as if they are machines doing a single task, 5) being vague about vagueness, 6) abstracting [away] from the things that really matter [to stakeholders].
- Karagiannis (2015) calls for “overcoming tendencies to view diagrammatic modeling methods and languages” as “stable, even standardized, artefacts that establish some commonly agreed way of describing a ‘system under study,’ [which] implies that all stakeholders work on the same level of abstraction and specificity.”
- Wyk and Heimdahl (2009) calls for introducing much greater flexibility in model-based development. “Based on years of experience with model-based development and formal modeling”, the authors report that “no single modeling notation will suit all, or even most, modeling needs” (p. 203) and ask for “building extensible and flexible modeling language processing tools” (p. 204).

Related research on modeling method usage (e. g., Fettke (2009) and Mendling et al. (2010)) and model comprehension (e. g., Haisjackl et al. (2018), Johannsen et al. (2014), and Mendling et al. (2019)) illuminates major issues. Many modelers do not apply modeling methods as intended by their designers, frequently using only a subset of the syntactic concepts provided (Langer et al. 2014). Modeling methods often do not fit modelers’ aptitudes, knowledge, and purposes (Hinkel et al. 2016; Zur Muehlen and Recker 2013). Simões et al. (2018) notes that the “lack of intuitiveness of diagrammatic representations and the complementary role of text-based representations has been underlined in recent research.” Cognitive load (Sweller 1994) for stakeholders becomes increasingly important as unfamiliar symbols and icons proliferate. Simões et al. (2018) also mentions lack of flexibility in process models, dilemmas of control, and excessive prescriptiveness. Uncertainty and variability related to accidents, mistakes, and intentional workarounds bring further challenges for modeling methods.

Requirements for a More Flexible Modeling Method. This paper pursues four requirements by presenting a modeling method that relaxes many common assumptions that are obstacles related to modeling and modeling methods.

1. The modeling method should respect stakeholder diversity related to knowledge, beliefs, and roles, thereby making it usable both by business professionals working individually and in collaboration and by IT professionals pursuing model-driven development or code generation. (cf. Fettke (2009)).
2. A modeling method can include different modeling techniques for different stakeholder purposes related to the same situation (contrary to the view in Karagiannis and Kühn (2002) that a modeling method can have only one modeling technique that combines a single modeling language and a modeling procedure).
3. With different modeling techniques for different purposes, a modeling method can use different modeling languages based on different metamodels. In relation to domain-specific conceptual modeling (cf., Karagiannis et al. (2016)), this approach assumes that intersubjective understanding between stakeholders might not require a single metamodel for processes, services, enterprises, goals, and so on.

4. The representation of a model might or might not use diagrams with rigorously defined notation and syntax (e. g., BPMN, ArchiMate) or might use such diagrams for some purposes but not for others.

Acceptance of multiple techniques, modeling languages, and metamodels within a modeling method leads to challenges related to maintaining coherence across different models produced by different stakeholders for different purposes. Our approach to coherence within a modeling method is to require use of a single overarching metaphor that applies to all modeling techniques within the modeling method. According to Ferstl and Sinz (2013, p. 138), a modeling method metaphor defines a specific perspective taken by the modeler while observing the reality and mapping the relevant aspects to the modeling language at hand, thereby creating a model representation of the reality. Ideally that metaphor should help in bridging gaps between modelers and practitioners who often visualize situations from different viewpoints. The invariance of the single overarching metaphor ensures that all modeling techniques contribute to an overall goal, even if they employ different levels of detail and expressiveness. Differences between the models will be revealed in stakeholders' personal understandings and collaborative discussions.

A Work System Modeling Method. This paper extends several decades of effort related to the work system method (WSM), a flexible systems analysis and design approach based on an informal type of modeling and problem-solving designed to help business professionals visualize work systems and collaborate more effectively with IT professionals (Alter 1995, 2006, 2013; Truex et al. 2010). Many hundreds of MBA and Executive MBA students have used various versions of WSM outlines that guided their production of management briefings about improving real world work systems. Those outlines contain many modeling techniques that have never been expressed as a formal modeling language (cf. Bork and Fill (2014)). For example, none is based on an explicit metamodel or operationalized in terms of a procedure.

Research Goal and Organization. This paper proposes a new, broader notion of modeling methods that enables greater flexibility. Central concepts within this notion include modeling method design spaces and metamodel design options. Those concepts are explained in relation to realization of a work system modeling method (WSMM) based on WSM, work system theory (WST), and a central work system metaphor. WST and WSM provide a plausible starting point for developing WSMM because their spirit is aligned with the "modeling for the masses" vision in Sandkuhl et al. (2018) and because an enterprise is a set of interacting work systems.

This paper provides contributions in several areas by building on Bork and Alter (2018) and Alter and Bork (2019). It introduces and explains a flexible notion of modeling method. It demonstrates that approach by showing that WSMM could help modelers and users apply a range of modeling techniques to situations that seem difficult to address without relaxing assumptions such as use of a single modeling technique, a formal modeling language, and diagrammatic notation. WSMM is a step toward a modeling tool that can be implemented using existing metamodeling platforms such as ADOxx. In a broader sense, it might serve as the theoretical basis for developing a modeling and analysis toolkit that could be used by business and IT professionals for individual purposes and to support collaboration. It might contribute to reflection on how modeling standards such as BPMN, EPC, and ArchiMate can be adapted to address needs of broader audiences.

The next section summarizes WST and WSM to introduce the central work system metaphor. A two-dimensional design space for flexible modeling methods illustrates design characteristics of modeling methods and metamodel design options. The design space is applied in describing a flexible WSMM modeling method traversing seven purposes of WSMM modeling that require a range of modeling techniques from quite informal to highly formal (cf. Bork and Fill (2014)). The subsequent section uses a hiring example to illustrate WSMM. A final section summarizes how WSMM addresses the requirements mentioned above, explains how WSMM maintains coherence across models built for different purposes, and identifies challenges for future research.

2 Foundations of Work System Theory and Work System Method

This section provides background related to the work system method (WSM), an informal modeling approach that to date has not been guided by a metamodel and that does not produce specifications in the sense of enterprise modeling.

Work system basics. A work system is a system in which human participants and/or machines perform processes and activities using information, technology, and other resources to produce product/services for internal and/or external customers. The "and/or" in the definition implies that work systems can be sociotechnical (with human participants) or totally automated. A work system operates within an environment that matters (e. g., national and organizational culture, policies, history, competitive situation, demographics, technological change, other stakeholders, and so on). Work systems rely on human, informational, and technical infrastructure that is shared with other work systems. Work systems should support enterprise and departmental strategies. The definition of work system implies that work system is a very general case that includes many special cases such as information systems, supply chains, service systems, projects, and totally automated work systems. For example, an IS is a work system most of whose activities are devoted to processing information. Supply chains are work systems that extend across multiple organizations to provide resources for other organizations. Projects are work systems that produce specific product/ services and then go out of existence. An enterprise or organization is a set of interacting work systems.

WST, the theoretical basis of WSM, consists of three parts: 1) the definition of work system,

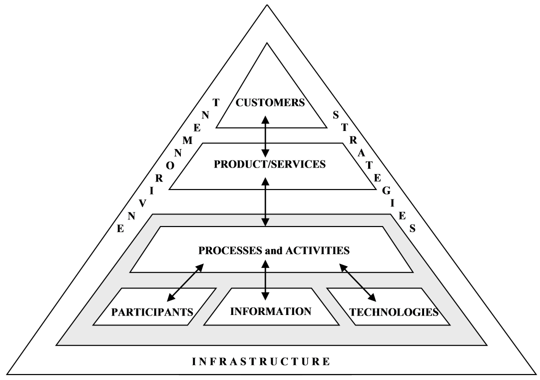

Figure 1: Work system framework (Alter 2006, 2013)

2) the work system framework, and 3) the work system life cycle model, which is not discussed here. This paper makes direct use of the definition and of the work system framework (Fig. 1), which outlines elements of even a rudimentary understanding of a work system's form, function, and environment as the work system exists during a time interval when its structure is basically static. Emphasizing business rather than IT concerns, this framework covers situations that might not have a well-defined business process and might not be IT-intensive. Processes and activities, participants, information, and technologies are viewed as completely within the work system. Customers and product/services may be partially inside and partially outside because customers often participate in work systems. A common limit to modeling precision is that human participants in work systems may make errors and may pursue adaptations and workarounds instead of following prescribed procedures. Furthermore, processes fall along a dimension from unstructured to structured (Alter and Recker 2017), starting with largely unstructured creative processes (such as many design and management processes) that have no pre-specified sequence, may involve extensive iteration, and therefore are not amenable to detailed, high precision modeling.

Work system method. WSM is a semi-formal systems analysis and design approach that was developed over several decades to help business professionals visualize work systems in their own organizations and collaborate more effectively with IT professionals. To date, almost all use of WSM has applied work system analysis outlines that suggest how to proceed from aspects of a work system's structure and performance toward producing a preliminary recommendation about how to improve the work system. The outlines include some questions that require textual answers, others that require filling out formatted tables, and others that invite users to include swimlane diagrams, Pareto charts, or other diagrams if they have appropriate software.

While details differ, every version of WSM is organized around the following: 1) identify the smallest work system that has the problem or opportunity; 2) summarize the "as-is" work system using a work system snapshot (example in Fig. 6), a stylized one page summary; 3) evaluate work system operation using measures of performance, key incidents, social relations, and other factors; 4) drill down further as necessary; 5) propose changes by producing a work system snapshot of a proposed "to be" work system that will probably perform better; 6) describe likely performance improvements.

3 Design Space for Modeling Methods

We agree with the view in Sandkuhl et al. (2018) that "not all knowledge should be represented as a formal model" and that it is important to find "the right balance of representational forms," including formal and informal models. A clear discussion of this entire topic requires a foundation such as the framework in Karagiannis and Kühn (2002) by which a modeling method is a composition of a modeling language, modeling procedure, and mechanisms & algorithms. A modeling language, the backbone of a modeling method, is composed of three components: syntax (the concepts provided by a modeling language, including their valid combinations), semantics (the meaning of the concepts), and notation (the graphical representation of the concepts). The combination of a modeling language with a modeling procedure is referred to as a modeling technique.

A more flexible perspective on modeling methods. Facilitating modeling by diverse stakeholders calls for relaxing the requirement in Karagiannis and Kühn (2002) that all stakeholders and purposes must be accommodated using a single modeling technique, i. e. one modeling language and one modeling procedure. Relaxing that requirement avoids unnecessary cognitive overload that could result from mixing concepts from separate modeling techniques that address diverse stakeholder needs and purposes. Separating modeling techniques that are used for different purposes maintains overall expressiveness of the modeling method without requiring that all users need to attend to every concept.

An alternative, more flexible view of modeling methods starts with a modeling method design space. The ideas explained next are equally applicable to design spaces based on the work system metaphor or other central metaphors such as systems in general, sociotechnical systems, actor networks, activity theory, and viable systems. This more flexible view accommodates both informal modeling for communication and collaboration and more formal models required for automated execution. Thus, a challenge emerges to find the best "[...] ratio between the machine language (strict and fixed metamodel) and the stakeholder aptitude" (Zarwin et al. 2012). The design characteristics presented next exemplify design options of method engineers aiming to provide flexibility in a modeling method.

3.1 Design Characteristics of Modeling Methods

Tab. 1 identifies five modeling method design characteristics related to the components of modeling methods proposed by Karagiannis and Kühn (2002): syntax, semantics, notation, modeling procedure, and mechanism & algorithms. The rows indicate the range of possibilities for each characteristic. In essence, the rows identify design choices for method engineers designing modeling methods. The discussion of these characteristics is based partly on a comparison of six enterprise modeling methods by Bork and Fill (2014) and

Table 1: Design characteristics of modeling methods

Design characteristic	**Range of design possibilities**		
Syntactic Expressiveness	Very small metamodels, e. g., < 10 concepts	Moderate number of syntactic concepts	Extensive number of syntactic concepts, e. g., > 100
Semantic Formality	No formal foundation, e. g., Visio stencils	Semi-formal foundation using natural language	Formal semantics using algebra, or ontologies
Visual Expressiveness	Natural language text	Tabular, outline-based, or matrix-based formats	Diagrammatic representation of conceptual models
Procedural Flexibility	Improvisational, i. e., without predefined steps to be performed	Semi-structured in sequence and content	Highly structured in sequence and content
Processing Capability	No model processing, i. e., models only serve visual representation means	Rudimentary processing, e. g., validation and queries	Complex processing, e. g., by means of simulations or interoperability with other tools
	Low <<< Technique specificity >>> High		
	High <<< Ease of use without extensive training >>> Low		

partly on our knowledge from analyzing more than 50 modeling methods within the Open Models Laboratory (OMiLAB) (Bork 2018; Bork et al. 2019).

The entries in the cells in Tab. 1 describe design dimensions using three levels that go from low to high. The bottom of Tab. 1 says that higher levels of all characteristics are associated with high method specificity and that lower levels tend to facilitate usage without extensive training. Those two variables are at the heart of many issues mentioned at the introduction. The high level of method specificity in rigorous modeling methods tends to make them difficult for non-experts, especially due to cognitive burdens of learning and applying formalisms and procedures, and executing complex model processing techniques. In contrast, experts rely on high levels of method specificity to assure soundness, validation, and consistency. Each of the characteristics will be discussed in turn.

3.1.1 Level of Syntactic Expressiveness

Expressiveness in conceptual modeling refers to the "degree to which a modeling language can describe all relevant aspects of a modeling domain" (Fettke 2009, p. 578). Thus, syntactic expressiveness increases with the number of concepts provided by a modeling technique's metamodel. A comprehensive metamodel covers all relevant syntactic concepts of the domain, while a narrow metamodel might only consider a single syntactic concept with a reflexive relation, e. g., an organizational metamodel comprised of only one concept 'Department' and one relation 'subordinated to'. A comprehensive enterprise modeling metamodel typically includes many more concepts. For example, the Multi-Perspective Enterprise Modeling (Frank 2014) metamodel has 208 concepts and the Semantic Object Model metamodel (Ferstl and Sinz 2013) has 105 (Bork 2018). Also, the widely used standard modeling languages are increasing in expressiveness with each new version. For example, the 110 concepts of UML version 1.1 in 1997 increased to 214 metamodel concepts in UML 2.0 (Ma et al. 2013). Similarly, the BPMN metamodel increased from 22 concepts in version 1.0 to 159 (an increase of 722%) in BPMN 2.0 (cf. Henderson-Sellers et al. (2012)).

This tendency toward increasing syntactic expressiveness has negative impacts on learnability and comprehensibility (Fettke 2009) of a modeling language. Furthermore, many users apply only a subset of the available concepts (Langer et al. 2014; Zur Muehlen and Recker 2013). The idea of alternative modeling techniques within one modeling method might encourage modeling method engineers to include techniques involving only the concepts necessary for a specific class of stakeholders needs. For example, a simple BPMN modeling technique might comprise only the concepts event, activity, sequence flow, and gateway. That technique seems feasible for describing the basic internal processes of an enterprise on a condensed level. In contrast, the development of workflow systems requires far more concepts, possibly all of the event types in the current version of BPMN, in order to model an execution environment precisely.

3.1.2 Level of Semantic Formality

Semantic formality is the extent to which statements in a modeling language are precise enough to be executed by a computer. People manage to conduct their everyday lives by communicating through natural language even though it has a very low level of semantic formality. Diagramming approaches such as BPMN have a higher level of semantic formality because they specify the logical progression of process flows, but they are too ambiguous to be executed automatically by machines. The highest levels of semantic formality involve languages such as Petri nets (Reisig 2012) that can be executed automatically by computers due to their formal dynamic semantics. Semantic formality thus refers to "the degree to which expressions in the language make precise statements" (Harel and Rumpe 2000, p. 18)

Increasing semantic formality enables further operationalization of the models, but at the cost of requiring extensive training for users. There is no rule of thumb for deciding which level of formality is best, and a high level of formality may not be better. Our new design space for modeling methods acknowledges this fact by recognizing that different stakeholder purposes often require different levels of formality. Thus, the semantic formality of different modeling techniques may differ significantly to accommodate different purposes of different stakeholders.

3.1.3 Level of Visual Expressiveness

This dimension refers to different ways of visually encoding conceptual models. Following the Physics of Notation (Moody 2009), visual expressiveness is measured by the number of visual variables (Bertin 1983) used in a modeling language notation. Using one obligatory visual representation creates unnecessary constraints on stakeholders who have different purposes. Much research focuses on establishing alternative or secondary notations (Ghiran et al. 2018), a concept also present in the latest version of de-facto standard modeling languages such as BPMN and ArchiMate (Bork et al. 2020). Different levels of visual expressiveness ease the comprehension of models and foster ease of use by means of an intuitive identification of the modeling concepts while at the same time serving the intended purpose of the modeling technique.

The use of alternative notations for any single concept within a modeling technique leads to construct redundancy, which generally impedes model comprehension (cf. Moody (2009)). In the context of a modeling method's design characteristics, however, we consider different notations on a larger scale, e. g., applying textual, tabular, matrix, and diagrammatic representations that are used in specific modeling techniques. Cognitive Fit theory says that different representations of information are suitable for different tasks and different audiences. Graphical symbols in software engineering should be understandable by both business and technical experts. Unfortunately, however, optimizing notations for novices can reduce their effectiveness for experts ("expertise reversal effect”). Moody (2009) suggests creating an “expert” and a “novice” version. In this regard, the positive aspects of alternative notation can be utilized while ensuring that modelers working with one modeling technique are not confused by alternative notations.

3.1.4 Procedural Flexibility

A modeling procedure specifies the way modelers need to utilize a modeling language to create and process valid models (Karagiannis and Kühn 2002). Procedural flexibility enables the alignment of the procedure to specific needs and purposes of stakeholders. A stringent procedure might be obligatory when formal models are used for simulation or transformation. On the other hand, the use of specific modeling tools rather than pen and paper constrains the modeling procedure. Recker et al. (2012) showed that people who perform work processes are commonly highly knowledgeable in the domain but lack methodological expertise in conceptual modeling.

Stringent procedures are frequently ignored in conceptual modeling methods even though they can help novices learn and use modeling methods. Introducing flexibility allows a given modeling method to provide different types of procedures for different users. That flexibility addresses an issue raised by Fettke (2009, p. 583) who states that “unified application of the modeling method can result in inadequate considerations of the requirements of a particular modeling problem.” Especially at the beginning of an analysis, procedural guidance might be quite informal, such as posting and organizing “post it” notes on a whiteboard, a common activity in participatory enterprise modeling (Stirna and Persson 2018) and in “design thinking,” where people produce informal models as part of a design method. Such early designs provide value, and sometimes can be transformed into diagrammatic forms for further processing (Miron et al. 2018).

3.1.5 Processing Capability

“Mechanisms provide the functionality to use and evaluate the models built by using the modeling language.” (Karagiannis and Kühn 2002, p. 186) Historically, modeling methods focused primarily on producing an adequate visual representation of a domain abstraction. A recent research note however argues that the value of conceptual modeling methods - and that of models - can be significantly increased when “amplifying the role of models from supporting communication and understanding towards the role of a machine-processable knowledge structure on which various mechanisms can be built” (Strecker et al. 2019, p. 244).

The design space for this characteristic ranges from no processing support at all, to moderate processing mechanisms, to the provision of highly-complex mechanisms & algorithms. An example of moderate processing of models is the automated syntactic and semantic validation of models, which requires almost no expert knowledge on the user side and is easy to utilize. An example of complex processing is model simulation, which requires extensive efforts on the modeler side, e. g., in creating the models along with specifying simulation parameters and configurations. Thus, model simulation requires expert knowledge about simulation data and simulation techniques plus conformance with precise procedures during the creation and preparation of the models.

3.2 Metamodel Design

The previous section discussed the design characteristics of modeling techniques with a focus on purpose and rationale. Next, we explain how these techniques are realized by means of metamodels that can be designed using any of the following three options: *integrated metamodels*, *independent metamodels*, or *interlinked metamodels*.

3.2.1 Integrated Metamodels

This metamodel design option provides tight integration of the metamodels for the different modeling techniques within the modeling method. That integration can be realized on the *meta-metamodel layer* or on the *metamodel layer*. Both options will be introduced.

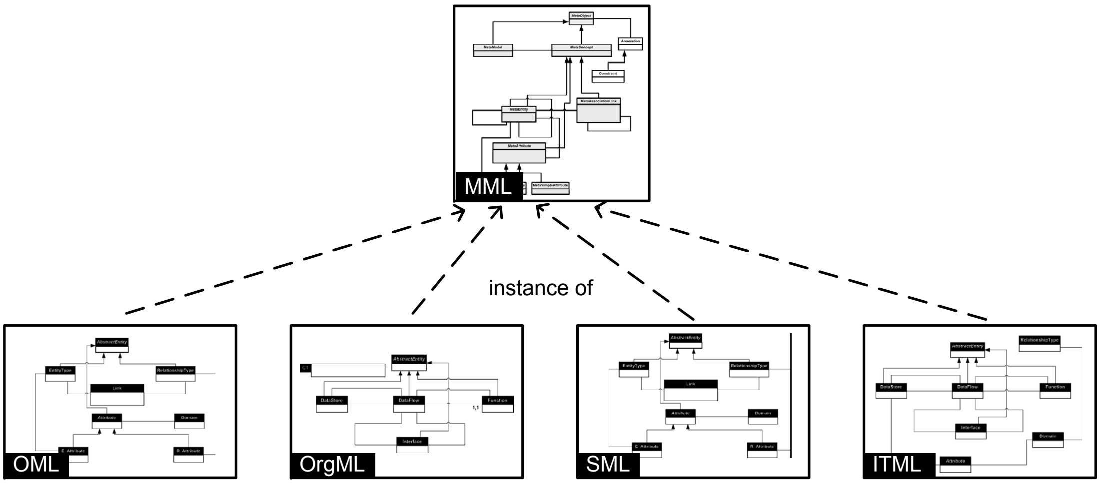

Figure 2: Integrated meta-metamodel of MEMO (Bock and Frank 2016) (adapted)

Integration on the Meta-metamodel Layer
With this approach, one overarching meta-metamodel applies throughout the entire modeling method. The different metamodels for selected modeling techniques are instances of the overarching meta-metamodel, i. e., each concept in each modeling technique metamodel is an instance of a concept provided by the meta-metamodel. One prominent example of this option is the Unified Modeling Language (UML) whose 13 different diagram types (i. e., modeling techniques) have independent metamodels albeit sharing the MetaObject Facility (MOF) as an overarching meta-metamodel. Examples of this type of integration from the enterprise modeling domain include the Multi-Perspective Enterprise Modeling (MEMO) approach (Frank 2014) that comes with the MEMO meta-metamodel (Frank 2014, p. 949) and the Semantic Object Model (SOM) approach that also comes with its own meta-metamodel (Ferstl and Sinz 2013, p. 141).

Fig. 2 illustrates the language architecture of the MEMO family of languages. At the top of the figure, the MEMO Metamodeling Language (MML) is depicted, forming the linguisitc foundation of MEMO (Bock and Frank 2016). All domain-specific languages such as OrgML and ITML are derived by instantiations of meta-concepts of MEMO MML. By following this approach, any domain-specific MEMO language shares meta-relationships with all other MEMO languages.

Integration on the Metamodel Layer
In contrast to the previous option, this option achieves integration by using a single overarching metamodel. The metamodels of the different modeling techniques are then defined by selecting relevant concepts from the metamodel (cf. metamodel slicing (Bork et al. 2020)). This option is exemplified in the SOM business process modeling metamodel shown in Fig. 3. The SOM method specifies one metamodel for business processes which is decomposed into four modeling techniques, each of which supports specific purposes. The metamodels of the four modeling techniques are slices of the overarching metamodel (Bork et al. 2020).

Integration on the metamodel layer also applies to ArchiMate (The Open Group 2017), the defacto industry standard for enterprise architecture management. ArchiMate provides an overarching metamodel from which the metamodels of the different ArchiMate layers are derived. It also uses a metamodel slicing technique. Bork et al. (2020) provides a detailed discussion of the ArchiMate metamodel.

3.2.2 Independent Metamodels

With this option, metamodels of different modeling techniques are completely independent from

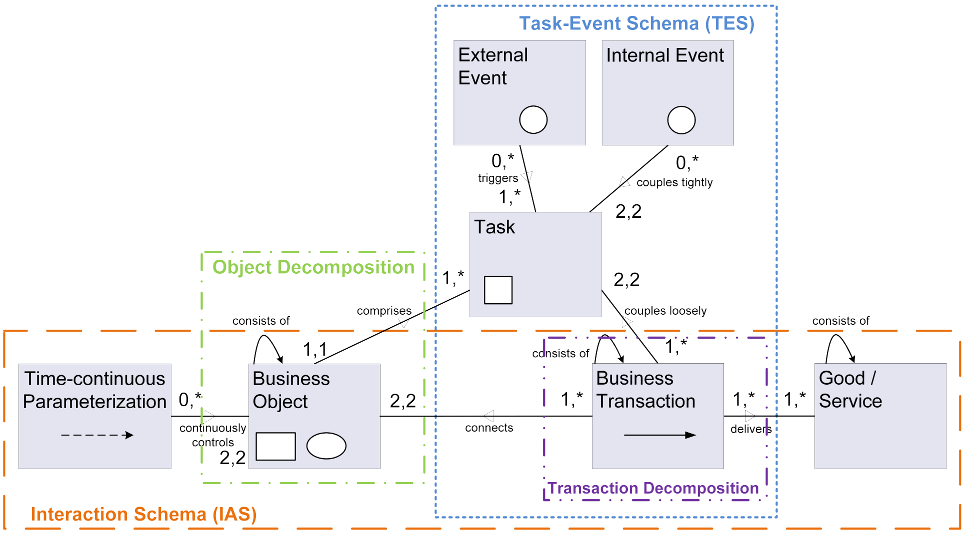

Figure 3: Integrated metamodel for SOM business processes ((Awadid et al. 2018), adapted (Ferstl and Sinz 2013))

each other. Thus, they neither share any overarching meta-metamodel nor are concepts of the different metamodels related to each other. The Zachman framework (Zachman 1987) for enterprise architecture management exemplifies this approach. Zachman decomposes the specification of an enterprise into a 6x6 matrix, each of which utilizing a specific representation technique (i. e., modeling technique). A key feature of the framework is that it explicitly omits specifying either modeling languages for each of the cells or procedures for applying the framework. That allows creation of models for each cell (or selected cells) using whatever modeling language seems most suitable.

Models based on the Zachman framework may have value as visual representations, but otherwise are quite limited. The limitation results from omitting a more formal and comprehensive specification of which languages to use and how the results of each Zachman framework cell are related to the other cells.

3.2.3 Interlinked Metamodels

The third metamodel design option is interlinked metamodels, whereby different metamodels are independent from each other, i. e., there exists neither an overarching meta-metamodel nor an overarching metamodel. Instead, loose coupling between the metamodels of the different modeling techniques is established by providing semantic linkages between selected concepts in the metamodels.

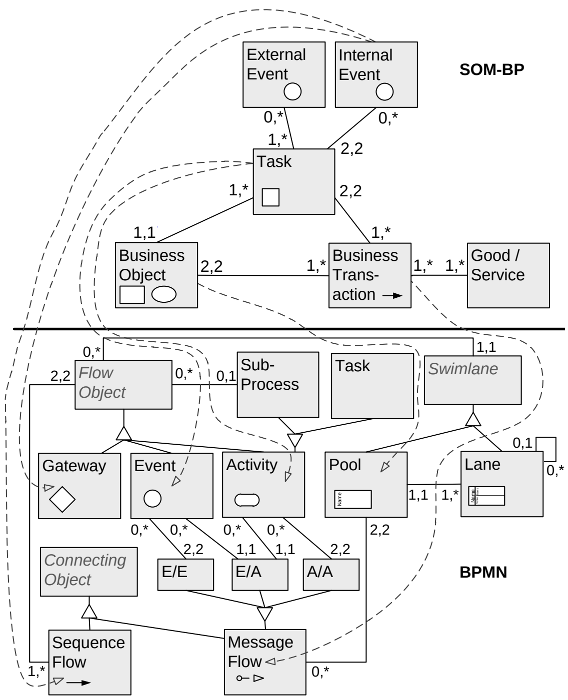

Figure 4: Interlinked metamodels between SOM and BPMN (Pütz and Sinz 2010, p. 67) (adapted)

Table 2: Summary of Metamodel Design Options

Design option	Strengths	Weaknesses
Integrated Metamodels	• Tight integration of all techniques • No syntactic inconsistencies • Extensions only affect one artifact	• Complex overarching metamodel • All changes need to be valid globally, i. e., for all techniques
Independent Metamodels	• Efficient development • No side-effects • Efficient extensions and update	• No syntactic coupling between different techniques • One technique at a time
Interlinked Metamodels	• Efficient development • Different techniques can be coupled, e. g., along a more complex modeling procedure	• Bidirectional transformations, effortful when new techniques are added • Side-effects (moderate)

Interlinked metamodels are exemplified between the SOM business process metamodel and the BPMN metamodel in Fig. 4. Integration of these two isolated metamodels is realized by semantic linkages. The semantic linkages moreover serve as a basis for model-driven development of workflow specifications from SOM business process models. Pütz and Sinz (2010) present a detailed description of this integration option.

3.2.4 Summary

The three metamodel design options provide different means of realizing coherence on the semantic level and integration on the syntactical level. Method engineers need to decide which type of integration is most suitable for the method at hand. Independent metamodels are more flexible, whereas integrated metamodels allow specifying an integrated model which leads to more powerful analysis possibilities.

Tab. 2 highlights strengths and weaknesses of the three metamodel design options. *Integrated metamodels* provide the strongest integration and do not introduce syntactic redundancies because they use a single metamodel. Moreover, extensions and updates of metamodel concepts only affect a single metamodel. A shortcoming of this design option is a tendency for a single metamodel to become too large as all metamodel concepts for different purposes are incorporated into the same metamodel. Additionally, all extensions and updates to the metamodel need to be globally valid, i. e., valid with respect to all modeling techniques.

Independent metamodels focus on a clear separation of concerns. Every technique comes with its own metamodel. Therefore, extensions and updates can be applied to the independent metamodels, which improves efficiency and flexibility. At the same time, independent metamodels have no syntactic coupling and naturally follow a 'one technique at a time' paradigm.

The design option *interlinked metamodels* is a compromise between the two previous design options of full integration versus no integration. Here, efficient development and revision of the metamodels is supported while side-effects are mostly handled by linkages. A weakness of this option is scaling with respect to the number of linkages. Strong coupling of the various metamodels leads to extensive effort in specifying and managing all linkages.

Having multiple metamodels that describe the same system under the umbrella of one modeling method inevitably introduces requirements for consistency. Those requirements challenge all of the design options, but in different ways. Models created with different techniques are likely to be related to each other because they are all part of the same modeling effort and specify parts of a larger system under study. These relationships should be managed by both the method engineer and the developer of corresponding modeling tools. Although beyond this paper's scope, future research should investigate the consequences of metamodel design options on consistency, perceived usefulness, and development efforts related to modeling tools or toolsets.

4 Work System Modeling Method

Fig. 5 represents the design space for the Work System Modeling Method. A key goal of the design space is accommodating a range of stakeholder purposes, shown as P1 through P7, all of which are related to a core metaphor. The shaded area represents the positioning of most of the modeling techniques that WSM users have applied. Most of those techniques focus on topics such as work system scope and operation, and activity/resource dependencies. Those techniques are relatively low in specificity compared to techniques that might be used for high precision description, system simulation, or code generation (cf. (Zarwin et al. 2012)). Instead of accepting the assumption that WSMM would include only one technique, this paper assumes that WSMM could include techniques anywhere in the design space provided that those techniques genuinely fit with its overarching metaphor.

The techniques included in WSMM can be based on an existing modeling language or on a modeling language designed specifically for WSMM. Modeling techniques that fit a modeling method's overarching metaphor and that address stakeholder purposes should be used instead of reinventing similar techniques. Thus, there are substantial benefits of using BPMN to describe processes or Entity Relationship diagrams to describe data. Sect. 3.2 identified techniques for integrating those modeling languages.

The WSMM design space shown in Fig. 5 should not be misunderstood as a linear sequence of purposes, where each subsequent purpose extends the previous one in some way. In contrast, the broad notion of a modeling method enables pursuing independent purposes such as specifying a system's data and specifying a system's strategy as long as all purposes reflect the same metaphor. The approach therefore addresses a problem in language engineering that *"typically overemphasise the challenges of mechanical manipulation of models, and neglect the variety of contexts, users and purposes for which models need to be created"* (Bjeković et al. 2014).

In the following, Sect. 4.1 presents a series of modeling techniques included in the Work System Modeling Method (WSMM) by sequentially following the purposes P1 through P7 as shown in Fig. 5. For each purpose, the rationale of the specific modeling technique is described and a sample model is shown. Sect. 4.2 presents metamodels for the modeling techniques associated with P1 through P6. This description emphasizes how the design options in Sect. 3.2 can be utilized in the context of WSMM to establish a heterogeneous but integrated view of work systems.

4.1 WSMM Modeling Techniques

WSMM expands WSM greatly by recognizing a wide range of purposes and different degrees of specificity for different stakeholders that all use the central work system metaphor. WSMM provides a broader scope of modeling to help various users understand the situation at hand and to decide how to improve it. WSMM uses simpler metamodels for informal and intuitive visualization of work systems and more precise and expressive metamodels for helping decision makers identify and select among possible changes and for helping software developers produce or

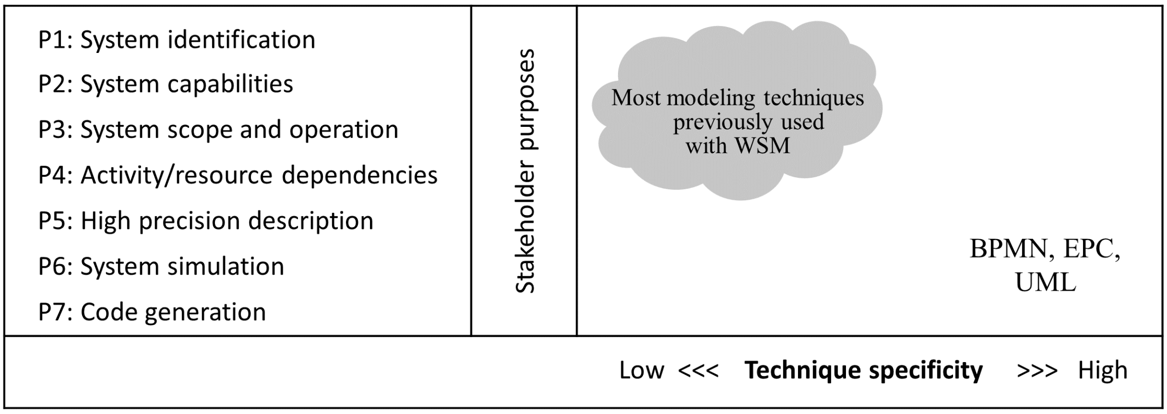

Figure 5: Work System Modeling Method Design Space

Table 3: Two illustrations of lists of capabilities (P2) for the hiring example

Simple list of capabilities	List of capabilities with performance or service level expectations
• defining parameters of the position • publicizing the position • prioritizing applications	• defining parameters for a position, responding within 3 days of request • publicizing the position, advertising in at least five bulletin boards • prioritizing applications, responding within three days of due date

improve software (cf. Zarwin et al. (2012)). The discussion uses the following hiring example to illustrate the scope of an initial version of WSMM in relation to purposes P1 through P7, but not to show all imaginable modeling techniques that might be included eventually.

Managers of an engineering firm are concerned that their current work system for finding, interviewing, and hiring job applicants takes too long, wastes too much effort in interviews, and seems to hire too many engineers whose contributions to engineering projects are disappointing. Some managers believe that a better online human resources (HR) portal would help. Others believe that the problems lie elsewhere.

4.1.1 Identification of the work system (P1)

Simply naming the work system of interest with a verb phrase such as “finding, interviewing, and hiring applicants” often avoids confusion with people thinking that the technology (e. g., the online HR portal) is the primary object of the improvement effort. Clarity in that regard makes sure that the project is viewed as much more than a software project. P1 makes no attempt to describe the work system’s behavior and structure and does not call for a specific procedure. The metamodel consists of one concept: Work System.

4.1.2 Capabilities of the work system (P2)

The hiring work system has capabilities that can be used as a service catalog elaboration of the P1 work system description. Those capabilities might be described using a list of verb phrases such as those in Tab. 3 (left column). A slightly more complete description of capabilities might include performance or service level expectations for each capability in Tab. 3 (right column).

A minimalist metamodel for P2 includes the concepts Work System and Capability. A slightly more detailed metamodel would add Performance expectation or Service level. The P2 capabilities lists in Tab. 3 are simple in form. Again, a specific

Vol. 15, No. 3 (2020). DOI:10.18417/emisa.15.3

Customers	Product/Services
• Applicants • Hiring manager • Larger organization (new colleague) • HR manager (who will analyze the nature of applications)	• Applications (which may be used for subsequent analysis) • Job offers • Rejection letters • Hiring of the applicant

Major Processes and Activities	
• **Hiring manager** submits request for new hire within existing budget • **Staffing coordinator** defines the parameters of the new position. • **Staffing coordinator** publicizes position. • **Applicants** submit job applications. • **Staffing coordinator** selects shortlisted applicants. • **Hiring manager** identifies applicants to interview.	• **Staffing coordinator** sets up interviews. • **Hiring manager** and **other interviewers** perform interviews. • **Hiring manager** and **other interviewers** provide feedback from the interviews. • **Hiring manager** makes hiring decisions. • **Staffing assistant** sends offer letters or rejections. • Successful **applicant** accepts or rejects job offer or negotiates further.

Participants	Information		Technologies
• Hiring managers • Staffing coordinator • Applicants • Staffing assistant • Other interviewers	• Job requisition • Job description • Advertisements • Job applications • Cover letters • Resumes	• Short list of applicants • Information and impressions from the interviews • Job offers • Rejection letters	• HR portal for communicating with applicants • Word processor • Telephones • Email

Figure 6: Illustration related to P3: A work system snapshot of a hiring work system

procedure is not employed. Researchers focusing on capability driven-development have used a more rigorous notion of capability, e. g., the 15 concepts in the capability-related metamodel in Zdravkovic et al. (2014).

4.1.3 Scope and general operation of the work system (P3)

With P3, the stakeholder wants to clarify the scope of the work system (based on P1 or P2) and to attain an overview of its general operation without going into great detail. Fig. 6 illustrates P3 in the form of a "work system snapshot", a modeling technique for describing a work system's scope and general operation. This is a formatted one-page summary of a work system in terms of the six central elements of the work system framework. Those six elements provide an easily used description that helps in defining the boundaries and contents of the work system that has the problem or opportunity at hand.

Work system snapshots increase syntactic expressiveness through a metamodel with these concepts: Customer, Product/Service, Activity, Participant, Information, and Technology. Cardinalities in the metamodel express internal consistency rules, e. g., each product/service must be received and used by at least one customer. The one-page tabular representation in Fig. 6 helps in visualizing a work system's scope by focusing on its core components. There is no need for a modeling procedure for producing this type of table.

4.1.4 Resources used and produced by specific activities (P4)

While a work system snapshot such as Fig. 6 is useful for discussing a work system's purpose and scope, many stakeholders need a deeper understanding of which resources are used and produced by each activity. Tables in the form of Tab. 4 are more useful for clarifying operational details by listing selected activities of a work system snapshot along with selected types of resources that are used and/or produced by those activities.

Various metamodels can be the basis of Tab. 4. A minimalist metamodel for P4 would include Work System, Activity, and Resource. It would treat all but the first column in Tab. 4 as resources that are used by the activities. Saying that actors are "used" by activities may sound strange, but it fits with the way some managers use the term resource in planning and management. For modeling, this provides a symmetrical way to handle human, informational, and technological resources. A more expressive metamodel might include all of the column headings in Tab. 4. An even more expressive metamodel presented in Alter and Bolloju (2016) contains over 50 concepts that identify other associations between activities and resources, such as which business rules affect customer-facing activities. Those increasingly elaborate metamodels move toward the level of specificity that programmers need, e. g., in identifying resources that are used or produced.

4.1.5 High precision description of the work system (P5)

Existing diagrammatic modeling techniques address many typical needs for understanding how the work system components are structured and how the work system operates. For example, it is easy to represent activity sequence and branching logic using BPMN diagrams with activities in swimlanes for different participant roles (see Fig. 7, which also applies to P6). While analysts might prefer a full version of BPMN, business stakeholders might prefer a simpler, restricted version of BPMN based on a minimalist metamodel whose concepts are limited to Swimlane, Start Event, Intermediate Event, End Event, Activity, Sequence Flow, and Message Flow (cf. Zur Muehlen and Recker (2013)). That would suffice for diagramming activities in Tables 3 and 4 even though it would require implicit handling of branching logic for situations such as when an applicant is rejected. Similarly, Entity Relationship diagrams and ArchiMate's application and technology layers could be used for P5 level descriptions of data structures and interactions between hardware and software. Thus, a high precision description of the work system needs to use modeling techniques that are not directly associated with WSM or WST but would be important to include in WSMM if it is to address needs of P5, P6, and P7.

For the sake of brevity, the process model in Fig. 7 only shows a version of the hiring process with a positive outcome. Details concerning the rejection of applicants or how the interviews are conducted are not included. Thus, every P4 activity specified in Tab. 4 is represented as a Task in the P5 BPMN model (Fig. 7).

4.1.6 Simulation of the work system's key processes (P6)

Some stakeholders may want to execute simulations to support deeper analysis of the "as-is" work system and deciding on the best of several possible future "to-be" work systems. For example, it might be useful to simulate the workload of different actors depending on the number of applicants, number of interviewers, and other factors.

A simulation model could apply an extended BPMN metamodel that adds simulation-specific concepts such as Statistical Distribution, and Random Generators, plus attributes such as transition conditions, probabilities, quantity, cost, and time. The green elements in Fig. 7 illustrate how those concepts can be added to a BPMN model for the hiring example. For example, probabilities need to be defined on the number of applicants being shortlisted and interviewed.

4.1.7 Code generation (P7)

Fig. 5 mentioned code generation because it is a central concern of many researchers in the modeling community. WSMM supports the effort to create understandings and artifacts that are needed in model-driven development but does not attempt to bridge the final gap between understandings and code.

Table 4: Illustration of P4: Selected resources used by a subset of Fig. 6 activities

Activities	**Actors**	**Information used, created, updated, or deleted**	**Technology**	**Trigger**	**Preconditions**	**Post-conditions and product/services produced**
Submit request for new hire.	Hiring manager	Hiring budget, Job requisition	HR portal	Need for new employee	Sufficient hiring budget	Job requisition exists
Define parameters of the job.	Staffing coordinator	Job requisition, Job description, Hiring policies	Word processor, HR portal	Job requisition	Job requisition	Job description
Publicize the job opening	Staffing coordinator	Experience with advertising media, Advertisement	HR portal, Web site for selected media	Job requisition, Job description	Job requisition, Job description	Advertisement displayed on websites
Submit application	Applicant	Job description, Cover letter, Job application, Resume	HR portal	Advertisement displayed on websites	Advertisement displayed on websites	Receipt of cover letter, job application, and resume
Select shortlist	Staffing coordinator	Job application, Short list of best applicants	HR portal	Deadline for job applications	Availability of job applications	Short list available to hiring manager
Identify applicants to interview	Hiring manager	Short list of best applicants, List selected for interviews	HR portal	Short list available to hiring manager	Short list available to hiring manager	List selected for interviews
Setup interview	Staffing coordinator	List selected for interviews, Schedule of interviews	HR portal	List selected for interview available	List selected for interview available	Schedule of interviews
Perform interview	Hiring manager	Job application, Schedule of interviews, Interview report	Word processor	Scheduled interviews start	Schedule of interviews	Interview report

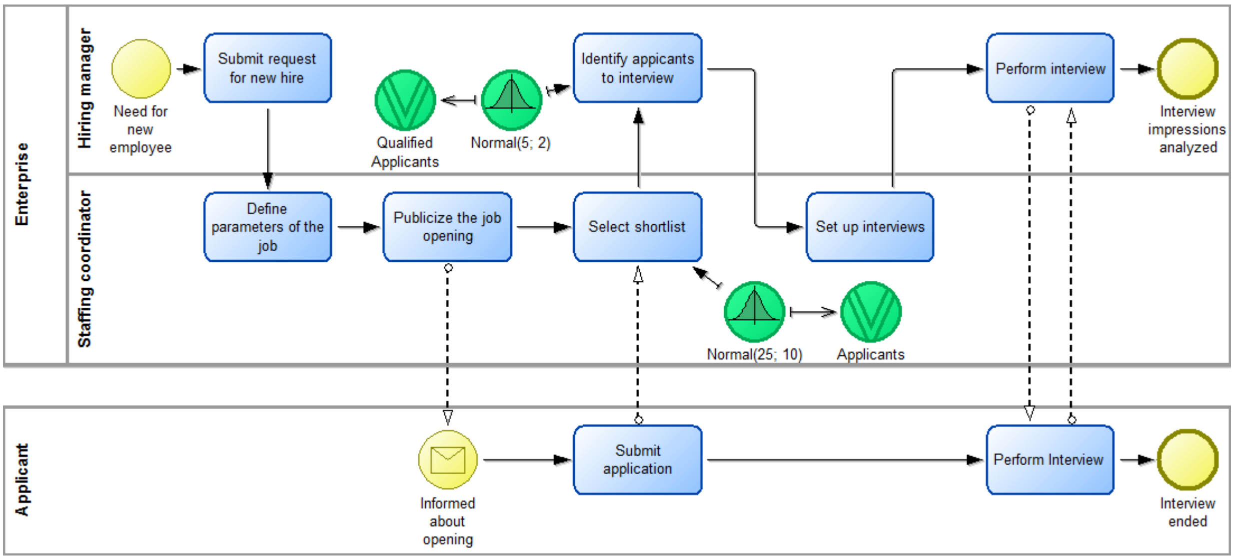

Figure 7: Part of a model supporting P5 (high precision description) and extended with the green elements to support P6 (simulation) of the hiring process

4.2 WSMM Metamodels

In the following, we will elaborate on potential metamodel designs (see Sect. 3.2) for purposes P1 to P6 (cf. Fig. 5). Code generation (P7) is not included because code generation is highly context-specific and dependent on the supporting code execution platform. Moreover, the purpose of generating code from a conceptual model does not necessary imply any changes to the underlying metamodel.

Fig. 8 shows potential metamodels that support the purposes P1 to P6 that were presented in the previous section. Object types are indicated by rectangles with gray color whereas connector types (i. e., classes that are instantiated as connectors relating two object type instances) are represented as non-directed arcs with italicized labels (cf. the metamodel specification technique proposed in Bork et al. (2020)). Cardinalities are further specified in min..max notation to constrain the allowed combinations between object types and connector types.

The following discussion concentrates on the metamodel design options applied in WSMM based on ideas from Sect. 4.1. Some relationships between elements of different modeling techniques are represented by the blue dashed lines labeled with 'Rn' in Fig. 8. Those relationships will be discussed as examples illustrating that WSMM utilizes all three metamodel design options mentioned in Sect. 3.2.

The metamodels for P1 and P2 share the same concept "Work System". WSMM applies an *integrated metamodel* approach by means of the P1 metamodel being a subset of the P2 metamodel (R1 in Fig. 8). The same applies to the P5 and P6 metamodels. The P6 metamodel extends the P5 metamodel by adding two object types ("Statistical Distribution" and "Random Generator") and two connector types (between "Activity" and "Statistical Distribution" and between "Statistical Distribution" and "Random Generator").

WSMM employs an *independent metamodel* approach in moving from P2 to P3 modeling techniques. Here, a very different abstraction level is employed by the two modeling techniques. The two metamodels do not overlap, i. e., there is no common metamodel element. However, P3 can be considered as a decomposition of the P2 models by specifying activities and resources required to provide certain products/services to a customer.

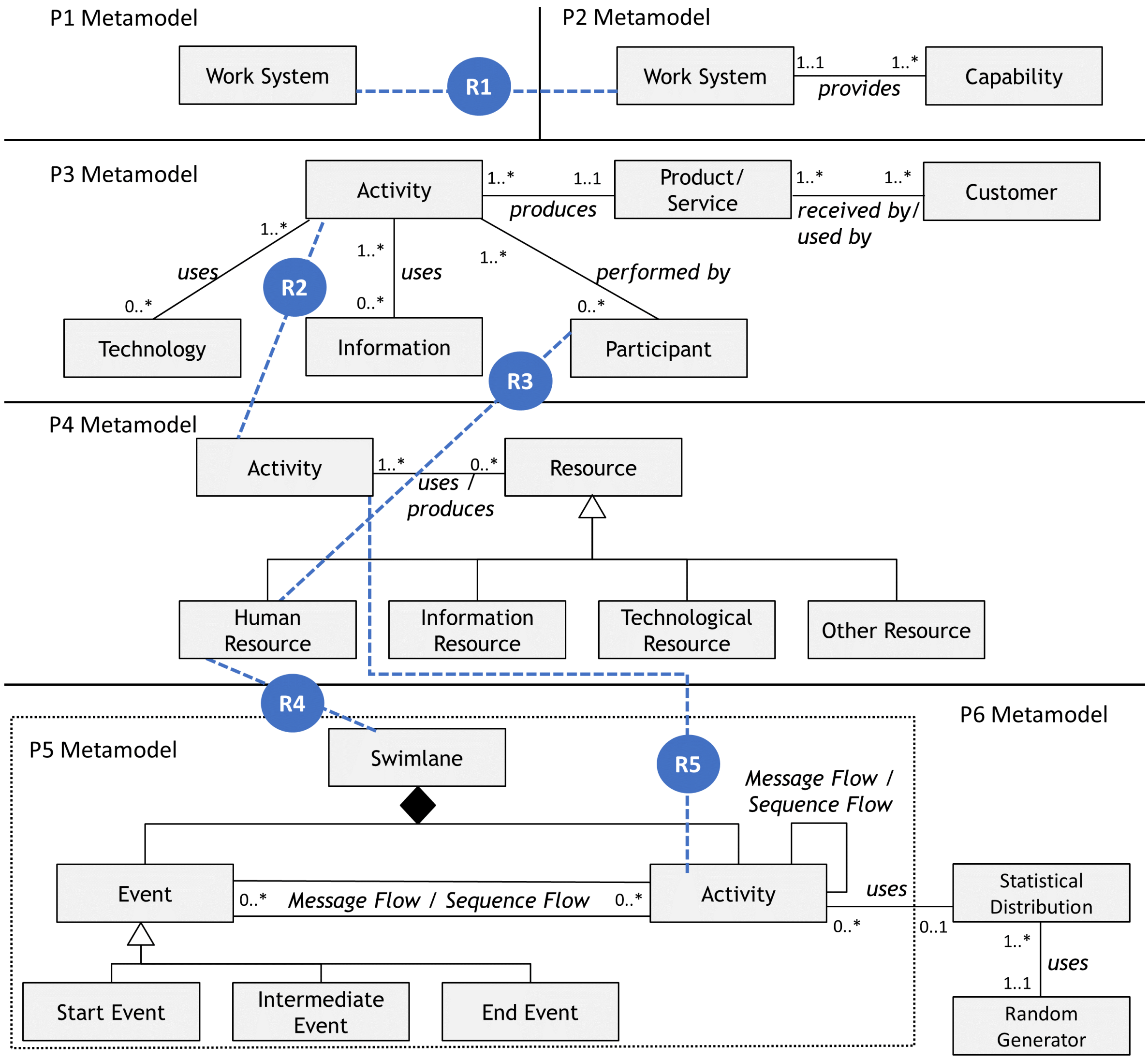

Figure 8: Potential WSMM Metamodels and applied Metamodel Design options supporting Purposes P1 to P6

Interlinked Metamodels is WSMM's most central metamodel design option. Relationships R2, R3, R4, and R5 in Fig. 8 exemplify the linkages between interlinked WSMM metamodels. Activities introduced in P3 are linked to P4 and also P5 activities (R2 and R5, respectively). Moreover, participants performing an activity in P3 are linked to human resources in P4, and eventually to swimlanes in P5. These relationships can be also visualized by following the participants of the P3 representation of the hiring example (Fig. 6), i. e., the hiring manager and staffing coordinator are also part of the P4 model (Tab. 4), and eventually are represented as swimlanes in the P5 BPMN model (Fig. 7).

5 Discussion

This paper was inspired by recently published concerns of leading researchers regarding current limitations of enterprise and process modeling. Its goal was to: i) propose an extended notion of a modeling method that accommodates more than one modeling technique in order to overcome some of the published concerns, and ii) exemplify the utilization of this extended notion by characterizing a practical modeling method that applies a central work system metaphor and can be used by stakeholders with different purposes and different levels of technical expertise. WSMM's approach to modeling methods relaxes widely accepted assumptions concerning the nature of modeling methods. That approach positions modeling methods and modeling techniques in a design space that traverses seven types of purposes that had not been articulated in that manner previously and that assumes different levels of technique specificity might be applied for any of those purposes.

We conclude with three topics: whether the four requirements for WSMM were met, how coherence is maintained, and areas for future research.

5.1 Satisfying Four Requirements for More Flexible Modeling Methods

WSMM exemplifies how the four requirements identified at the outset can be realized. 1) In contrast to most modeling methods, WSMM respects stakeholder diversity by recognizing that different stakeholder purposes may generate different needs for expressiveness, rigor, and completeness. 2) WSMM includes multiple modeling techniques, as shown in the previous sections. 3) WSMM can use multiple modeling languages based on different metamodels, but all related to the same work system metaphor. 4) WSMM includes diagrams where needed by stakeholders, e. g., for P5 and P6, but does not require diagrams for other purposes such as P2, P3, and P4.

WSMM also addresses other issues mentioned at the outset. WSMM conforms with the willingness in Sandkuhl et al. (2018) to soften requirements for completeness, coherence, and rigor to achieve broader and more effective usage of modeling. WSMM addresses six of seven process modeling problems noted by Aalst (2012): 1) WSMM assumes that different users with different purposes will prefer models with different characteristics; 2) WSMM does not straitjacket processes into one monolithic model; 3) WSMM does not use static decomposition; 4) WSMM does not treat people as machines doing a single task; 5) WSMM is clear about vagueness in its recognition of unstructured and semi-structured processes, and adaptations; 6) WSMM is designed to help people improve work systems and does not abstract away from things that matter, such as performance. In relation to issues from (Karagiannis 2015), WSMM assumes that stakeholders will work at different levels of abstraction and specificity and that their needs may not be satisfied by a single modeling language.

5.2 Coherence within WSMM

The challenge of coherence in WSMM can be viewed as making sure that WSMM is more than just an assemblage of techniques. At minimum, the use of metamodels that all build on the same core metaphor (the work system) should facilitate production of consistent models for different purposes. For example, a P2 model of capabilities should create a logical constraint on a P3 model that summarizes the work system scope and operation that enacts those capabilities. P3 processes and activities become the basis for the activities in P4 models, and so on. This paper proposes different metamodel design options, reflecting different extents of integration amongst the modeling techniques as practical guidance for addressing coherence. Future research needs to investigate the perceived usefulness and ease of use of these design options.

A larger question is whether WSMM would help business professionals understand work systems for their own purposes while also helping them collaborate with IT professionals with different world views. The use of different metamodels developed for different purposes reduces the likelihood of an automatic way to zoom between the various modeling techniques. However, the progression of the different stakeholder purposes (P1 through P7) outlines a path for communicating between stakeholders who have different purposes. P1 names the work system in the form of the verb phrase. All stakeholders should be able to rally around creating or improving that work system.

P2 and P3 support relatively informal understandings of the work system's content, capabilities, and scope. A software engineer in a work system improvement project needs to understand levels P1, P2, or P3 to communicate effectively with business professionals about topics other than isolated details, and also to make it less likely that software will miss basic issues. P4 covers details that business professionals need to verify and that IT professionals need to understand to produce software. Both business and IT professionals might have their own needs for delving into detailed process models, data models, or technical interface models created for P5. In all cases, knowledge of models for P1 through P4 will help in understanding both details and significance of models for P5. Only specialists will pursue P6 and P7, and it would be difficult for them to do that well without understanding models for P3 and P4.

Much of coherence across the various modeling techniques in WSMM needs to be achieved through people looking at models and discussing ambiguities and questions with other stakeholders. Much of the coherence is in the quality of the models and their application of work systems as a common modeling metaphor. The rest of the coherence for P1 through P5 is in the minds and collaborative spirit of the stakeholders, i. e., not just in documentation, and it depends partly on the quality of interaction with other stakeholders.

While the different techniques are based on the same metaphor, they can be applied independently and in whatever order the stakeholders prefer. The sequence from P1 to P7 was presented in a linear way to explain the ideas, but users might want to proceed in an iterative manner. For example, they might start with a first cut at P3 and then might go back to P1 and P2 to clarify how they want to name and bound the work system and how they want to articulate its capabilities. Most of the recent attention to design thinking and agile development is consistent with a mindset of being clear about what is being done, but relying heavily on iteration across interim steps and not worrying about performing work in a linear sequence except in situations where that is important for producing a good result.

5.3 Areas for Future Research

WSMM could help modelers and users apply a range of valuable modeling techniques to situations that seem difficult to address convincingly without relaxing widely held assumptions about the strict relationship between a modeling method and a modeling language. Here are some of the research opportunities that represent next steps:

Exploring general implications of the WSMM vision. This paper showed how relaxing common assumptions about modeling methods enables comprehensive visualization of complex systems such as work systems. We do not know where synergies and conflicts between this view and more established views might take us. Challenging, extending, and/or validating these ideas requires iterations of discussion, feedback, and possibly revision.

Implementation of a widely accessible WSMM toolkit. This paper focused on explaining ideas and concepts rather than illustrating technical implementations. In a parallel effort we demonstrated technical feasibility of implementing WSMM by using ADOxx,[1] a widely used metamodeling development and configuration platform for implementing modeling methods that is available through OMiLAB (the Open Models Laboratory), an open community for the conceptualization of modeling methods (Bork et al. 2019). ADOxx provides capabilities that can be used to implement all of the metamodels mentioned in this paper plus all of the modeling techniques found in existing Microsoft Word analysis outlines used by MBA and EMBA students in recent years. ADOxx also supports other work system modeling techniques such as those related to conformance to sociotechnical principles, anticipation of workarounds, customer responsibilities for specific work system activities, and the value of product/services to specific customer groups. Further work and experience with

[1] ADOxx metamodeling platform,http://www.adoxx.org, last visited: 31.07.2019

Table 5: Modeling, analysis, and design modules of a WSMM toolkit

Modeling modules	**Analysis modules**	**Design modules**
• Identification • Capabilities • Operation and scope of the work system • Value capture • Responsibilities • Visibility • Activity/resource dependencies • System interactions • Diagrammatic specifications	• Performance gaps • Strengths and weaknesses • Exceptions • Workarounds or noncompliance • Key incidents • Risks • Issues for elements of the work system framework	• Proposed changes in the work system • Rationale for proposed changes • Likely improvements in work system performance

ADOxx is required to identify the most convenient ways to use its modeling capabilities across the many topics in existing WSM outlines and other topics based on new metamodels.

Preliminary attempts to identify components of a WSMM toolkit based on ADOxx led to realizing that the toolkit would be much more valuable if it contained modules that included analysis and design in addition to modeling. Tab. 5 identifies WSMM modules for modeling, analysis, and design.

Replication for other test cases. Many aspects of this paper, such as the design characteristics of modeling methods (Tab. 1) and the metamodel design options (Tab. 2) superimposed a rigorous modeling viewpoint on top of useful but less rigorous ideas associated with WSM as it has existed to date. That exercise of describing a flexible modeling method based on an overarching metaphor should be attempted for other sets of ideas, such as general systems theory, sociotechnical theory, actor network theory, and soft system methodology. That would require experts in those areas of theory and practice to identify modeling techniques that are used or could be used, to specify underlying metamodels, and to explore the possibility of producing modeling methods that researchers and practitioners in those areas would find useful.

Further development of the research stream related to WSM, WST, and extensions. The development of WSM started several decades ago with a focus on issues related to P3 and with little or no attempt at rigor other than trying to define terms and encourage organized thinking about work systems that involved IT. The ideas that became WSM and WST evolved gradually.

The current research extends that stream of research in many directions. It overcomes the limiting assumption that the research was mostly centered around what Fig. 6 would call a P3 analysis by business professionals. It eliminates an outdated assumption of a single work system metamodel that needs to be highly detailed. The new approach is potentially much more valuable because it calls for many alternative metamodels based on the same central metaphor but designed for different purposes.

6 Conclusions

This paper introduced a new vision of modeling methods by which a modeling method may include a variety of modeling techniques that are related to an overarching metaphor but that may address different purposes and with different levels of syntactic expressiveness, semantic formality, visual expressiveness, procedural flexibility, and processing capability. It showed how widely accepted restrictions concerning the nature of modeling methods such as a single modeling technique and formal, diagrammatic models could be overcome with a new approach that positions modeling methods and modeling techniques in a design space that traverses different purposes. In addition, this paper proposed three metamodel design options method engineers can use in order to adopt the new modeling method notion: integrated metamodels, independent metamodels, and interlinked metamodels.

This paper demonstrated potential usefulness by applying that vision to a test case, a new Work System Modeling Method (WSMM) that comprises a variety of modeling techniques related to work systems. Examples illustrated selected WSMM modeling techniques that were based on separate metamodels that are all related to the same core metaphor. The test case also showed how the different metamodel design options could be utilized within a coherent modeling method. We hope to continue developing WSMM both because of its potential value in application and because it seems to be a useful test case for continuing development of modeling methods that are more flexible and more useful for multiple purposes and stakeholders.

References

van der Aalst W. M. (2012) What makes a good process model? In: Software & Systems Modeling 11(4), pp. 557–569

Alter S. (1995) How should business professionals analyze information systems for themselves? In: Information System Concepts: Toward a Consolidation of Views, Proceedings of the IFIP WG 8.1 Working Conference on Basic Information System Concepts, Marburg, Germany. Springer, pp. 284–299

Alter S. (2006) The work system method: Connecting people, processes, and IT for business results. Work System Press

Alter S. (2013) Work system theory: Overview of core concepts, extensions, and challenges for the future. In: Journal of the Association for Information Systems 14(2), pp. 72–121

Alter S., Bolloju N. (2016) A Work System Front End for Object-Oriented Analysis and Design. In: International Journal of Information Technologies and Systems Approach (IJITSA) 9(1), pp. 1–18

Alter S., Bork D. (2019) Work System Modeling Method with Different Levels of Specificity and Rigor for Different Stakeholder Purposes. In: 14th International Conference on Wirtschaftsinformatik (WI2019). University of Siegen, Germany / AISeL, pp. 124–138

Alter S., Recker J. C. (2017) Using a work system perspective to expand BPM use cases for research. In: Journal of Information Technology Theory and Application 18(1), pp. 47–71

Awadid A., Bork D., Karagiannis D., Nurcan S. (2018) Toward generic consistency patterns in multi-view enterprise modelling. In: Twenty-Sixth European Conference on Information Systems (ECIS2018), Portsmouth,UK, 2018. AISeL

Bertin J. (1983) Semiology of graphics: diagrams, networks, maps. University of Wisconsin press

Bjeković M., Proper H. A., Sottet J.-S. (2014) Embracing Pragmatics. In: International Conference on Conceptual Modeling. Lecture Notes in Computer Science Vol. 8824. Springer, pp. 431–444

Bock A., Frank U. (2016) Multi-perspective enterprise modeling—conceptual foundation and implementation with ADOxx. In: Domain-specific conceptual modeling. Springer, pp. 241–267

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. IEEE, pp. 3400–3409

Bork D. (2018) Metamodel-based Analysis of Domain-specific Conceptual Modeling Methods. In: IFIP Working Conference on The Practice of Enterprise Modeling. Springer, pp. 172–187

Bork D., Alter S. (2018) Relaxing Modeling Criteria to Produce Genuinely Flexible, Controllable, and Usable Enterprise Modeling Methods. In: Proceedings of the 9th International Workshop on Enterprise Modeling and Information Systems Architectures. CEUR-WS.org, pp. 46–50

Bork D., Buchmann R. A., Karagiannis D., Lee M., Miron E.-T. (2019) An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem. In: Communications of the Association for Information Systems 44, pp. 673–697

Bork D., Karagiannis D., Pittl B. (2020) A Survey of Modeling Language Specification Techniques. In: Information Systems 87, in press

Clark T., Frank U., Kulkarni V. (2016) Supporting Organizational Efficiency and Agility: Models, Languages and Software Systems https://www.dagstuhl.de/de/programm/kalender/semhp/?semnr=16192

Correia F. F., Aguiar A. (2013) Patterns of flexible modeling tools. In: Proceedings of the 20th Conference on Pattern Languages of Programs. ACM, pp. 1–9

Ferstl O. K., Sinz E. J. (2013) Grundlagen der Wirtschaftsinformatik. Oldenbourg Verlag, München

Fettke P. (2009) How conceptual modeling is used. In: Communications of the Association for Information Systems 25(1), pp. 571–592

Figl K. (2017) Comprehension of procedural visual business process models. In: Business & Information Systems Engineering 59(1), pp. 41–67

Frank U. (2014) Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. In: Software & Systems Modeling 13(3), pp. 941–962

Ghiran A.-M., Buchmann R. A., Karagiannis D. (2018) Towards a framework of techniques for enabling semantics-driven secondary notation in conceptual models. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS). IEEE, pp. 1–6

Gonzalez-Perez C. (2018) Vagueness. In: Information Modelling for Archaeology and Anthropology: Software Engineering Principles for Cultural Heritage. Springer, pp. 129–141

Haisjackl C., Soffer P., Lim S. Y., Weber B. (2018) How do humans inspect BPMN models: An exploratory study. In: Software & Systems Modeling 17(2), pp. 655–673

Harel D., Rumpe B. (2000) Modeling languages: Syntax, semantics and all that stuff - Part I: The Basic Stuff.. Technical report. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.1512

Henderson-Sellers B., Qureshi M., Gonzalez-Perez C. (2012) Towards an interoperable metamodel suite: Size assessment as one input. In: International Journal of Software and Informatics 6, pp. 111–124

Hinkel G., Kramer M., Burger E., Strittmatter M., Happe L. (2016) An empirical study on the perception of metamodel quality. In: 2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD). IEEE, pp. 145–152

Johannsen F., Leist S., Braunnagel D. (2014) Testing the impact of wand and weber's decomposition model on process model understandability. In: International Conference on Information Systems (ICIS) 2014. AISeL, pp. 1–13

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

Karagiannis D., Kühn H. (2002) Metamodelling Platforms. In: E-Commerce and Web Technologies, Third International Conference, EC-Web 2002, Aix-en-Provence, France, September 2-6, 2002, Proceedings. Springer, p. 182

Karagiannis D., Mayr H. C., Mylopoulos J. (2016) Domain-Specific Conceptual Modeling, Concepts, Methods and Tools. Springer

Langer P., Mayerhofer T., Wimmer M., Kappel G. (2014) On the usage of UML: Initial results of analyzing open UML models. In: Modellierung 2014. Gesellschaft für Informatik e.V., pp. 289–304

Ma Z., He X., Liu C. (2013) Assessing the quality of metamodels. In: Frontiers of Computer Science 7(4), pp. 558–570

Mendling J., Recker J., Reijers H. A., Leopold H. (2019) An empirical review of the connection between model viewer characteristics and the comprehension of conceptual process models. In: Information Systems Frontiers 21, pp. 1111–1135

Mendling J., Reijers H. A., van der Aalst W. M. (2010) Seven process modeling guidelines (7PMG). In: Information and Software Technology 52(2), pp. 127–136

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

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

Pütz C., Sinz E. J. (2010) Model-driven derivation of BPMN workflow schemata from SOM business process models. In: Enterprise Modelling and Information Systems Architectures (EMISAJ) 5(2), pp. 57–72

Recker J., Safrudin N., Rosemann M. (2012) How novices design business processes. In: Information Systems 37(6), pp. 557–573

Reisig W. (2012) Petri Nets: An Introduction Vol. 4. Springer

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. In: Business & Information Systems Engineering 60(1), pp. 69–80

Simões D., Antunes P., Carriço L. (2018) Eliciting and modeling business process stories. In: Business & Information Systems Engineering 60(2), pp. 115–132

Stirna J., Persson A. (2018) Enterprise Modeling: Facilitating the Process and the People. Springer

Strecker S., Baumöl U., Karagiannis D., Koschmider A., Snoeck M., Zarnekow R. (2019) Five Inspiring Course (Re-) Designs. In: Business & Information Systems Engineering 61(2), pp. 241–252

Sweller J. (1994) Cognitive load theory, learning difficulty, and instructional design. In: Learning and instruction 4(4), pp. 295–312

The Open Group (2017) ArchiMate Specification Version 3.0.1 https://publications.opengroup.org/c179

Truex D., Alter S., Long C. (2010) Systems Analysis for Everyone Else: Empowering Business Professionals through a Systems Analysis Method that fits their needs. In: Proceedings of the 18th European Conference on Information Systems. AISeL, p. 4

Wüest D., Seyff N., Glinz M. (2019) FlexiSketch: A lightweight sketching and metamodeling approach for end-users. In: Software & Systems Modeling 18, pp. 1513–1541

Wyk E. V., Heimdahl M. P. E. (2009) Flexibility in modeling languages and tools: A call to arms. In: Int J Softw Tools Technol Transfer 11(3), pp. 203–215

Zachman J. A. (1987) A framework for information systems architecture. In: IBM systems journal 26(3), pp. 276–292

Zarwin Z., Sottet J.-S., Favre J.-M. (2012) Natural modeling: Retrospective and perspectives an anthropological point of view. In: Proceedings of the 2012 Extreme Modeling Workshop. ACM, pp. 3–8

Zdravkovic J., Stirna J., Kuhr J.-C., Koç H. (2014) Requirements engineering for capability driven development. In: IFIP Working Conference on The Practice of Enterprise Modeling. Springer, pp. 193–207

Zur Muehlen M., Recker J. (2013) How much language is enough? Theoretical and practical use of the business process modeling notation. In: Seminal Contributions to Information Systems Engineering. Springer, pp. 429–443