Design Metamodels for Domain-Specific Modelling Methods using Conceptual Structures

Wilfrid Utz*

Department of Knowledge Engineering,
Faculty of Computer Science, University of Vienna,
Währingerstr. 29, 1090 Vienna, Austria
wilfrid.utz@univie.ac.at
https://informatik.univie.ac.at/ke/

Abstract. Enterprises nowadays operate in fast-changing environments and need to adapt dynamically to new circumstances. This impacts the way how enterprise information systems are analysed, designed and implemented. Conceptual modelling methods experience a constant evolution nowadays. The methods are re-constructed continuously to reflect the changing application/industrial domain. It is therefore required to provide additional design support to realise domain specific modelling methods and more precisely their underlying metamodel.
The goal of this research project is to enable the knowledge representation of a metamodel to support its design process. Building upon the assumption that conceptual structures within a metamodel exist, a knowledge representation framework is proposed using conceptual graphs as the mathematical baseline. The proposal will be evaluated in a laboratory setting, applied on metamodel design challenges observed.

Keywords: Metamodelling · Knowledge Representation · Conceptual Structure · Domain-specific Modelling Methods.

1 Introduction

Enterprises operate today in fast changing environments. External factors influence and disrupt their business models; technology advances at a rapid pace and changes the way organisations offer their products and services; legal/regulatory requirements need to be reviewed continuously and respected.

All these factors influence the way information systems are designed, analysed and used. Enterprise modelling methods are an established means for the conceptual analysis, design and implementation of complex enterprise systems. They models realised using these methods provide value as a documentation

*Supervisor: o. Univ.-Prof. Prof.h.c. Dr. Dimitris Karagiannis means and to support understanding by human beings[22], formalise the knowledge base and enable knowledge management and knowledge engineering[18].

In the field of conceptual modelling, we can distinguish between general purpose approaches, that aim to provide a standardised conceptual modelling foundation for a specific application field and domain-specific modelling methods that target specific interpretation and operation needs of an enterprise. The metamodel reflects these aspects at the core of any domain-specific modelling method: model-based operations define the concepts required to syntactically and semantically describe the organisation (or a subset of it), and enable model processing to create value.

Within this research project, domain-specific modelling methods are in focus, specifically targeting the challenge of designing "adequate" metamodel to support dynamic (in the sense of quickly evolving) and complex (vertically and horizontally integrated) environments. An initial understanding of "adequate" in relation with a domain-specific metamodels has been derived from [17, p. 5] where usefulness defines scope of the abstraction performed in the design process. Requirements derived during the design and runtime aspects influence the design process. As a solution proposal, it is intended to extend the notion of metamodel towards a conceptual structure. The term conceptual structure is understood and closely linked to the work performed by Sowa in [26] as a logic-based knowledge representation technique. Derived from linguistics, a conceptual structure allows the representation of concepts in terms of a small number of conceptual primitives [21] that can be expressed mathematically as a conceptual graph. The assumption in this research project is that conceptual structures in metamodels exist and can support the design process of domain-specific modelling methods.

2 Problem Description

Problem Observed. The design process for metamodels as the core element of a modelling method is observed as a challenging issue: the task to derive a consistent metamodel that is adequate for a specific domain and application, enables model value and provides interpretation support for stakeholders involved, requires the metamodel engineer requires conceptual, technological and domain-specific knowledge to base design decisions upon and weight in and compromise on different interests.

Observing current practices in the field of metamodelling, these knowledge challenges are introduced using the example of a development process observed graphically shown in Fig. 1.

1. *Understand requirements:* in order to develop an adequate and useful metamodel, the requirements for knowledge operations have to be captured and understood. Knowledge on the application and industrial domain is required to trigger the design process, e.g. simulation of manufacturing processes, dependency analysis of community networks, verification of deadlocks in

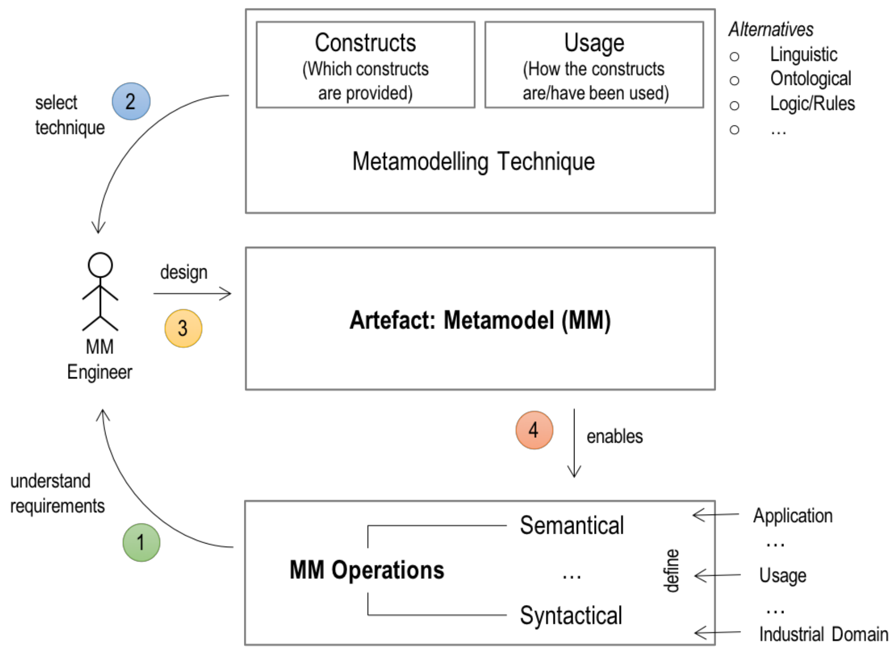

Fig. 1. Example: Metamodel Development Approach

automated business processes, integration of metamodel on different formalisation levels into a consistent state for enterprise architecture management in a given industrial domain.

2. *Select a meta-modelling technique:* based on these requirements, the appropriate meta-modelling technique has to be selected. The technique provides generic constructs to develop a metamodel, construction principle and usage patterns and consequently enables specific functionalities e.g. code generation, reasoning techniques on ontological metamodels, deduction in logic based environments. The functionalities offered have to be mapped to the requirement as a fundamental design decision that has to be taken. Knowledge on specific techniques applicable is needed.
3. *Design the metamodel:* having accomplished these 2 prerequisites, the metamodel is designed in conformance with the selected metamodelling approach. The design techniques are typically adjustment, mapping or re-use patterns and the resulting metamodel is strongly influenced by the expertise and creativity of the metamodel engineer in the overall approach, design decisions taken and specific usage techniques.
4. *Enable metamodel operations:* knowledge operations realise the value of models, enabled by the corresponding metamodel. These operations are established functionalities like composition (e.g. in case of sliced metamodel as discussed in [4]), binding of model processing algorithms through reference

alignment, transformation and semantic lifting [13] and operate potentially on the structure and semantic provided by the metamodel. This steps verifies the applicability of the metamodel.

These knowledge-based challenges strongly influence the efficiency to establish an adequate design of a metamodel. The selection of a meta-modelling technique is currently driven by the metamodel engineers knowledge on a specific approach rather than the requirements elicited; during the process adjustments are performed to overcome this initial selection issue. It is assumed that these adjustment have an impact on the usefulness of the resulting metamodel. As a description framework of specific, existing metamodels is not available, a re-use of results is limited. This results in a re-implementation of similiar structures (using the same or a different technique).

The motivational example below showcases the observation. Different meta-modelling techniques have been applied to represent the same metamodel requirement from a structural point of view: the sequence definition of a business process. The application of different approaches results in varying functionality based on the design and technique used.

Motivational Example: BPMN 2.0. The observation on the construction of meta-models and its dependency on the meta-modelling approach is visually shown in Fig. 5. The Business Process Model and Notation (BPMN) 2.0 [24] has been selected to exemplify the issue as this standard is extensively discussed in research in the past years. The example has been selected as it nicely demonstrates how different metamodelling techniques impact conceptualization results. Driven by required capabilities of the metamodel, the results differ even though the same concept is designed.

The representation shows the aspect of sequence flows (highlighted in red/dashed line) of the BPMN metamodel and how it has been designed using three different meta-modelling techniques: a) the formal specification from the BPMN 2.0 standard document using class diagrams [24, p. 144], b) an ontology-based approach to conceptualise the specification [25, p. 140] and c) a logic-based representation using rules mapped to petri net constructs (introduced in [15, p. 51] extended in [14]). The selected meta-modelling does not only impact the way concepts are represented, but is also related to the functionality required and enabled by the approach. For the example above, the following can be observed:

a) *Metamodel using UML Class Diagrams*: intends a formal representation that can be used to generate code as it can be mapped to elements of object-oriented programming languages or other execution systems. In line with the BPMN 2.0 specification, "execution semantics have been fully formalised" [24, p. 10] and allow conformance verification as well as runtime interpretation.
b) *Metamodel using OWL Representation*: using ontology concepts enables verification and checks of model artefacts e.g. by "checking the compliance of a

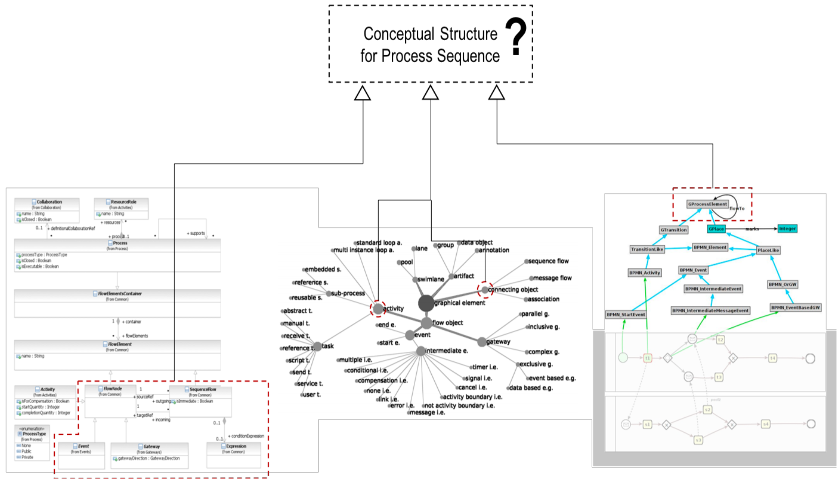

Fig. 2. BPMN 2.0 Process Metamodel represented as a) UML 2.0 Class Diagram, b) OWL Ontology, c) Logic/Rules using TELOS[23]

process against the BPMN specification" [25, p. 136], reasoning and detection mechanisms. The sequence logic is implemented as object properties in the ontology.

c) *Metamodel using TELOS*: using logic and rules that are bound to the petri net meta-structure provide simulation capabilities as the behaviour is described already in the abstract meta-model. The sequencing possibilities are represented as a self-referential loop on the root element.

It is assumed that the problem described using the motivational example is not only applicable for pre-existing metatmodels (and would result in an integration challenge of different realizations) but can be observed also for domain-specific metamodels and their characteristics that are constructed from scratch.

Gap. Resulting from the problem observed above, the design or (continuous) adaptation/evolutions of metamodels is a tedious, inefficient and error-prone task that requires domain expertise on one hand, meta-modelling knowledge, an overview of existing results and creativity to perform the adjustment and interpretation of requirements to a new/evolved metamodel. It is proposed to close this gap by a) extending the notion of metamodels as conceptual structures as a means to define and describe metamodels independent of the technique used and collect these conceptual structures for re-use in the engineering process of metamodels. The adequateness, usefulness and purpose of the metamodel might influence the need for expressiveness of the conceptual structure for metamodels.

The toolbox of a metamodel engineer should include this collection and evolve along industrial trends in a dynamic manner. Supporting agile techniques as discussed in [17] has been identified as a challenge this research project aims to contribute to as the means of re-use of metamodels, consistent combination/integration and evolution for "Models of Concepts" on the other hand (see Fig. 3 in [17, p. 8] are currently limited.

Research Objectives. Based on the identified gap, the research questions are formulated below. As a design based research methdology is followed (introduced in section 3. The objective of the research project is to develop a knowledge representation formalism as a conceptual structure for metamodels, representing the domain knowledge (syntax, notation, semantic, behaviour) encapsulated as elements in the structure. The design artefacts realized as the contribution are mapped to research objectives and questions depicted graphically in Fig. 3.

RQ1: *Which conceptual structure can be identified as adequate to support the formalization of metamodels?*

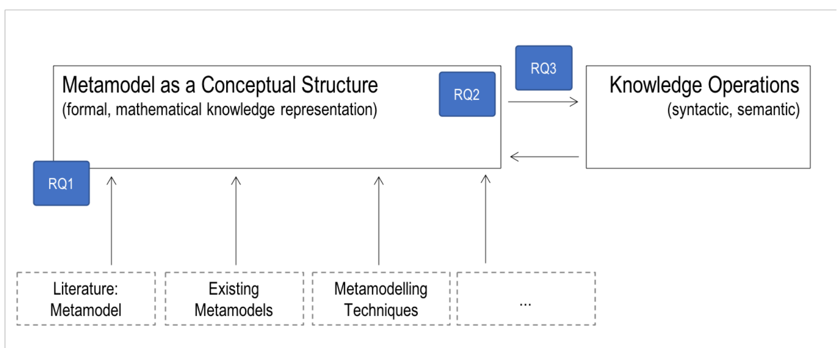

Fig. 3. Research Question: Metamodels as a Knowledge Representation

RQ1 aims to develop a framework for knowledge representation as conceptual structures that is adequate to describe metamodels. It is currently assumed that this knowledge representation can be derived from a) the definition of the term "metamodel" in the scientific community, b) existing metamodels from academia and industry and their characteristics and construction principles, c) meta-modelling techniques and patterns that are currently applied, and d) knowledge operation as requirements on metamodels. Based on these preliminary input identification, the objective of the research questions is to identify a formalism that allows the description of metamodels using dimensions to be established as part of the research questions (e.g. syntactical, semantical, behavioural, operational, etc.).

RQ2: *Which mechanisms are required to describe and collect conceptual structures of metamodels? How can reoccurring patterns in existing metamodels be identified and described in abstract terms?*

The research questions targets the mechanisms required to a) describe and b) collect/support re-use metamodels as conceptual structure. This includes the analysis of description dimensions that enable e.g. their registration, retrieval and binding. Having the motivational example in mind, the second part of the research question targets the identification of patterns based on these established dimensions. Metamodels developed using modelling techniques of similar or varying expressiveness and functionality are reviewed to identify techniques to describe patterns and provide means for e.g. generalisation/abstraction of these patterns or fragments independent.

RQ3: *Do algorithms exist that are appropriate to suggest and propose knowledge operations in a domain context using the conceptual structure?*

The focus of this research question is to identify how knowledge operations can be proposed for specific conceptual structures. This builds on the assumption that specific operations require a conceptual structure to operate upon. It is planned to investigate available algorithms that suggest functionality for metamodels based on expressiveness of the conceptual structure and functional requirements not limited to interoperability, semantic lifting, reference alignment, mapping or consistency management. The assumption underpinning the research question is that a metamodel needs to fulfil certain requirements to be capable to perform a concrete knowledge operation to be applicable.

3 Research Methodology

The research objective of this project is to develop artefacts to support the engineering of domain-specific modelling methods, more precisely the design of its metamodels. A novel and innovative framework that integrates the conceptual structures as a building block will be proposed that allows a knowledge representation of metamodels. The design-science based approach introduced by Hevner and Chatterjee in 2010 in [11] has been selected as the adequate research methodology for the project. As an initial step, the Information Systems Research Framework in [11, p. 274] has been specialised to the research project, respecting the guidelines set forth by Hevner and Chatterjee. The research methodology for the project is presented in Fig. 4.

In line with the research questions, the research project aims to develop an adequate conceptual structure to represent metamodels and corresponding matching algorithms for knowledge operations. The design of this formalism and algorithms is evaluated through a prototypical implementation of a domain-specific language for metamodels. As a means to iteratively refine the design

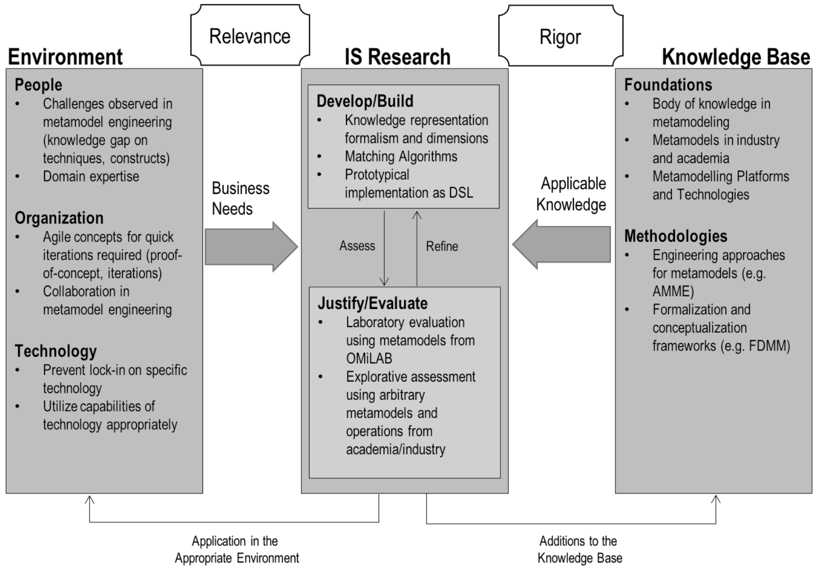

Fig. 4. Research Methodology derived from [11, p. 274]

artefacts, the concept established and prototype realised are continuously evaluated in the context of the Open Models Initiative Laboratory (OMiLAB) [9] using metamodels developed in the laboratory in a first phase (for a collection of domain-specific modelling methods developed in the context of the OMiLAB see [19]), extended to a broader scope from academia and industry thereafter. These evaluation iterations will refine the design artefact and prototypical implementation iteratively.

4 Research Approach and Preliminary Results

The research approach for the project is structured according to the research methodology. These phases contribute individually to the research questions and are used during the Assess-Refine iterations. The five phases are executed in line with the methodology and will be iteratively run. This impacts individual design-evaluation cycles of each phase and also transitions in between phases. In general, phase 1 and 2 are contributing to the explication and requirements elicitation objective of the project, phase 3 to a refinement, whereas phase 4 performs the development and phase 5 the evaluation of the design artefacts.

Phase 1: Analyse the Knowledge Base.
(contributing to RQ1 and RQ2)

During this phase, a literature review is conducted to establish an understanding of the terminology and definition applicable for metamodels and conceptual structures. The understanding of the term metamodel and its different interpretation will impact the design of the conceptual structure and vice-versa. The applicability of the research question is verified during this phase and further refined.

Preliminary Results. A preliminary review of literature on available definitions for the term meta-model aims to establish a common baseline. The results are presented below having motivated in an early stage of the research project the identification of the gap and research questions.

- *Language-based Understanding of Metamodels:* Strahinger investigates in [28] the term meta-model and how it is used in scientific literature. 24 selected definitions have been assessed, showing the ambiguity of the term and its application in scientific literature of the domain. The author concludes by constructing the term using a language-oriented technique, discussed in [27].
- *Ontological Metamodels:* The ontological representation of metamodels has been extensively researched in the past following the objective to enrich metamodels (type semantics) and models (inherent semantics) with more expressiveness e.g. to cover semantic interoperability requirements (see [12]). The ontological representation provides the required formal foundation for a domain ([35]).
- *Situative Metamodels:* Brinkkemper discusses in [5] situative metamodels. This aspect of meta-modelling aims to compose metamodels from fragments that support a specific subset of the modelling methods. The discussion of the language elements required to describe method fragments, including the ontological anchoring as defined in [6] is relevant for the conceptual structure definition in this research project. The extended view on situational methods in [10] supports the assessment.
- *Metamodelling Platforms:* Karagiannis/Kühn introduce [16] a generic framework for modelling methods and establish the definition of the metamodel on meta^2 level. The framework is composed of the model technique (consisting of the modelling procedure and modelling language) and the model processing functionalities as mechanisms and algorithms to be considered during the conceptualisation.
- *Metamodels in Model-Driven Development* Atkinson/Kühne review in [1] the types of metamodels used in Model-driven Development (MDD). They identify which abstraction techniques are currently used and applied by software engineers to write higher-level code: traditional, OMG based modelling infrastructures, linguistic metamodelling and ontological metamodelling. The representations resulting from the techniques are classified as views that conform to the technique applied.

The review of literature is in progress and conclusions are to be established. A review on the terminology and its application in the software engineering domain (e.g. model-driven development and meta-meta models in [20], graph-based techniques in [7], multi-level metamodelling in [8]) is pending. Additionally, the results developed by the conceptual structure community in computer science is reviewed and assessed.

An observation at this stage shows that the research questions are in line with the challenges observed in literature (e.g. [1] or [4]

Phase 2: Collect and Assess Metamodels.
(contributing to RQ1 and RQ2)

Concrete metamodels are identified and collected in this phase. This collection acts as a basis for further analysis on the formalisation concepts and dimensions. Fragmentation techniques as discussed in [4] from a specification viewpoint might act as guidelines to assess subsets and patterns within existing metamodels. The collection aims to develop classification dimensions in a first step and verify those continuously. As a second aspect, the knowledge operations that can be observed within metamodelling projects are collected and classified. The classifications scheme used is input for the dimensions used in the conceptual structure (see [36] for an example of a similar approach on knowledge management services).

Preliminary Results. The collection phase is in progress and builds at this stage on the metamodel developed at the OMiLAB and the literature review performed in [2].

Phase 3: Refinement of Requirements.
(contributing to RQ1 and RQ2)

The objective of this phase is to assess and evaluate requirements and the resulting development/evaluation results in iterations. It is responsible to capture intermediate results and feed the forward/back to the related phases.

Preliminary Results. Results from the design/evaluation iterations are not yet available. Preliminary work has been performed in research projects that fed into the design phases introduced above. The results achieved are published in [34], [33], [32] and [30]. The presented results in these publications has been done in differing industry and application domain, focusing on the development of metamodels in these fields.

Phase 4: Establish the Conceptual Structure and Realise a Prototypical Implementation
(contributing to RQ2 and RQ3)

The objective of this phase is to design the concept and implement a prototypical realisation for conceptual structures.

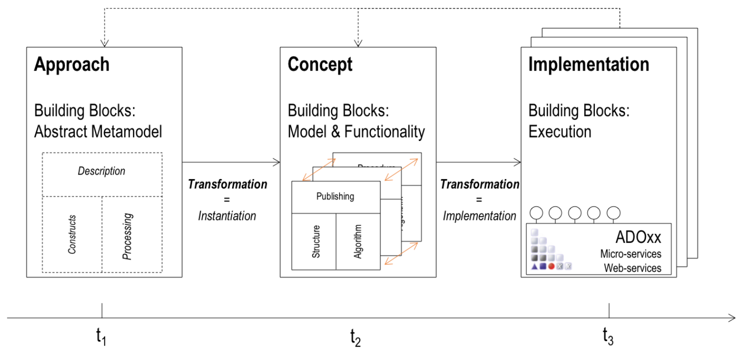

Fig. 5. Metamodelling using Abstract Building Blocks

Preliminary Results. As a preliminary result a procedural framework has been introduced using abstract metamodelling building blocks. This framework acts as a guiding element for the research to be conducted in the review and analysis phases. It has been proposed in [3] and prototypically been applied in [29].

1. *Approach:* at this stage, abstract metamodel building blocks are introduced. Each building block specified couples abstract model constructs, related model processing functionality and a description of the description of the abstract metamodel building block that enables the identification and integration of it in more complex scenarios that span multiple building blocks. The description should reflect the dimensions required for the conceptual structure.
2. *Concept:* the transformation from the approach stage is done through instantiation. The abstract blocks are made concrete using a selected metamodelling technique.
3. *Implementation:* represents the operationalisation level of the building blocks as the conceptual blocks are mapped to a concrete realisation technology and can be executed/run in a tooling infrastructure.

Phase 5: Evaluate and Refine Design Artefacts
(contributing to RQ1, RQ2 and RQ3)

The evaluation of the concept and prototype is performed using identified metamodels in an explorative laboratory setting. Feedback from the evaluation refines the concept developed.

Preliminary Results. Evaluation results in a structured form are not yet available. An initial indications for the applicability of the concept is discussed in [31] using reference alignment operation on two metamodels for for industrial business process management (IBPM) to support simulation mechanisms on a graph-based structure. The prototype developed and documented in [29].

Acknowledgment

This doctoral project is supervised by o. Univ.-Prof. Prof.h.c. Dr. Dimitris Karagiannis, head of the research group Knowledge Engineering and the Open Models Initiative Laboratory (OMiLAB) at the Faculty of Computer Science, University of Vienna.

References

1. Atkinson, C., Kuhne, T.: Model-driven development: a metamodeling foundation. IEEE Software **20**(5), 36–41 (Sept 2003). https://doi.org/10.1109/MS.2003.1231149
2. Bork, D.: Metamodel-based analysis of domain-specific conceptual modeling methods (in press). In: Proceedings of 11th IFIP WG 8.1 working conference on the Practice of Enterprise Modelling. PoEM '18 (2018)
3. Bork, D., Fill, H.G., Karagiannis, D., Utz, W.: Simulation of Multi-Stage Industrial Business Processes Using Metamodelling Building Blocks with ADOxx. Enterprise Modelling and Information Systems Architectures **13**, 333–344 (February 2018). https://doi.org/https://doi.org/10.18417/emisa.si.hcm, http://eprints.cs.univie.ac.at/5413/
4. Bork, D., Karagiannis, D., Pittl, B.: How are Metamodels Specified in Practice? Empirical Insights and Recommendations. In: Twenty-fourth Americas Conference on Information Systems (AMCIS 2018). pp. 1–10. Twenty-fourth Americas Conference on Information Systems, New Orleans, 2018 (August 2018), http://eprints.cs.univie.ac.at/5577/
5. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and software technology **38**(4), 275–280 (1996)
6. Brinkkemper, S., Saeki, M., Harmsen, F.: Meta-modelling based assembly techniques for situational method engineering. Information Systems **24**(3), 209 – 228 (1999). https://doi.org/https://doi.org/10.1016/S0306-4379(99)00016-2, http://www.sciencedirect.com/science/article/pii/S0306437999000162, 10th International Conference on Advanced Information Systems Engineering
7. Ebert, J., Franzke, A.: A declarative approach to graph based modeling. In: International Workshop on Graph-Theoretic Concepts in Computer Science. pp. 38–50. Springer (1994)
8. Frank, U.: Multilevel modeling. Business & Information Systems Engineering **6**(6), 319–337 (2014)
9. Goetzinger, D., Miron, E.T., Staffel, F.: OMiLAB: An Open Collaborative Environment for Modeling Method Engineering. In: Karagiannis et al. [19], pp. 55–76, http://dblp.uni-trier.de/db/books/collections/KMM2016.html

10. Henderson-Sellers, B., Ralyté, J.: Situational method engineering: state-of-the-art review. Journal of Universal Computer Science (2010)
11. Hevner, A., Chatterjee, S.: Design Research in Information Systems (2010)
12. Hoefferer, P.: Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In: sterle, H., Schelp, J., Winter, R. (eds.) ECIS. pp. 1620–1631. University of St. Gallen (2007), http://dblp.uni-trier.de/db/conf/ecis/ecis2007.html
13. Hrgovcic, V., Karagiannis, D., Woitsch, R.: Conceptual modeling of the organisational aspects for distributed applications: the semantic lifting approach. In: Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual. pp. 145–150. IEEE (2013)
14. Jeusfeld, M.: Semcheck - script for the case studies for summerschool (03 2017)
15. Jeusfeld, M.A.: Semcheck: Checking constraints for multi-perspective modeling languages. In: Karagiannis et al. [19], pp. 31–53, http://dblp.uni-trier.de/db/books/collections/KMM2016.html
16. Karagiannis, D., Kühn, H.: Metamodelling Platforms. Lecture Notes in Computer Science pp. 182–182 (2002)
17. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics. pp. 5–10. PCI '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2801948.2802040, http://doi.acm.org.uaccess.univie.ac.at/10.1145/2801948.2802040
18. Karagiannis, D., Buchmann, R., Walch, M.: How can Diagrammatic Conceptual Modelling Support Knowledge Management? In: 25th European Cenference on Information Systems, ECIS 2017. pp. 1568–1583. Proceedings of the 25th European Conference on Information Systems (ECIS) (June 2017)
19. Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.): Domain-Specific Conceptual Modeling, Concepts, Methods and Tools. Springer (2016), http://dblp.uni-trier.de/db/books/collections/KMM2016.html
20. Kern, H., Hummel, A., Kuehne, S.: Towards a comparative analysis of meta-metamodels. In: 11th Workshop in Domain-Specific Modeling (2011), http://www.dsmforum.org/events/DSM11/Papers/kern.pdf
21. Kerstens, J., Ruys, E., Zwarts, J.: Conceptual Structure. Lexicon of Lingustics (2001)
22. Mylopoulos, J.: Conceptual modelling and Telos. Conceptual Modelling, Databases, and CASE: an Integrated View of Information System Development, New York: John Wiley & Sons pp. 49–68 (1992)
23. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. ACM Trans. Inf. Syst. **8**(4), 325–362 (Oct 1990). https://doi.org/10.1145/102675.102676, http://doi.acm.org.uaccess.univie.ac.at/10.1145/102675.102676
24. OMG: Business Process Model and Notation (BPMN), Version 2.0 (January 2011), http://www.omg.org/spec/BPMN/2.0
25. Rospocher, M., Ghidini, C., Serafini, L.: An ontology for the business process modelling notation. In: Garbacz, P., Kutz, O. (eds.) FOIS. Frontiers in Artificial Intelligence and Applications, vol. 267, pp. 133–146. IOS Press (2014), http://dblp.uni-trier.de/db/conf/fois/fois2014.html
26. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)
27. Strahringer, S.: Ein sprachbasierter metamodellbegriff und seine verallgemeinerung durch das konzept des metaisierungsprinzips. In: Modellierung. vol. 98, pp. 15–20 (1998)

28. Strahringer, S., et al.: Zum Begriff des Metamodells. Techn. Hochsch., Inst. für Betriebswirtschaftslehre (1995)
29. Utz, W.: GO0D MAN Project, D4.2 ZDM Data and Management Environment Implementation (2018), http://go0dman-project.eu/wp-content/uploads/2018/05/Abstract-D4.2-ZDM-Data-and-Management-Environment-Implementation.pdf, to be published
30. Utz, W., Karagiannis, D.: Towards business and IT alignment in the future Internet: Managing complexity in e-business. In: Advances in Future Internet, 2009 First International Conference on. pp. 18–23. IEEE
31. Utz, W., Lee, M.: Industrial Business Process Management Using Adonis Towards a Modular Business Process Modelling Method for Zero-Defect-Manufacturing. 2017 International Conference on Industrial Engineering, Management Science and Application (ICIMSA) pp. 1–5 (2017)
32. Utz, W., Reimann, P., Karagiannis, D.: Capturing Learning Activities in Heterogeneous Environments: A Model-Based Approach for Data Marshalling. In: IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014, Athens, Greece, July 7-10, 2014. pp. 624–626. https://doi.org/10.1109/ICALT.2014.182, https://doi.org/10.1109/ICALT.2014.182
33. Utz, W., Woitsch, R.: A Model-Based Environment for Data Services: Energy-Aware Behavioral Triggering Using ADOxx. In: Collaboration in a Data-Rich World. pp. 265–275. IFIP Advances in Information and Communication Technology, Springer, Cham
34. Utz, W., Woitsch, R., Karagiannis, D.: Conceptualisation of Hybrid Service Models: An Open Models Approach. In: Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual. pp. 494–499. IEEE, http://ieeexplore.ieee.org/abstract/document/6032287/
35. Wand, Y.: Ontology as a foundation for meta-modelling and method engineering. Information and Software Technology **38**(4), 281 – 287 (1996). https://doi.org/https://doi.org/10.1016/0950-5849(95)01052-1, http://www.sciencedirect.com/science/article/pii/0950584995010521, method Engineering and Meta-Modelling
36. Woitsch, R., Karagiannis, D.: Process-oriented knowledge management systems based on KM-services: the PROMOTE approach. Intelligent Systems in Accounting, Finance and Management **11**(4), 253–267. https://doi.org/10.1002/isaf.235, https://onlinelibrary.wiley.com/doi/abs/10.1002/isaf.235