frontiers | Frontiers in Computer Science

TYPE Editorial
PUBLISHED 05 December 2023
DOI 10.3389/fcomp.2023.1326413

OPEN ACCESS

EDITED AND REVIEWED BY
Guiming Luo,
Tsinghua University, China

*CORRESPONDENCE
Heinrich C. Mayr
✉ heinrich.mayr@aau.at

RECEIVED 23 October 2023
ACCEPTED 22 November 2023
PUBLISHED 05 December 2023

CITATION
Mayr HC and Thalheim B (2023) Editorial: Model-centered software and system development. *Front. Comput. Sci.* 5:1326413. doi: 10.3389/fcomp.2023.1326413

Editorial: Model-centered software and system development

Heinrich C. Mayr[1]* and B. Thalheim[2]

[1]Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria, [2]Christian-Albrechts-Universität zu Kiel, Kiel, Germany

KEYWORDS

model centered architecture, conceptual modeling, IoT systems, model transformation, model semantics

Editorial on the Research Topic
Model-centered software and system development

Modeling is the key ability of humans to understand and master their environment. Accordingly, humans use models as instruments for managing complexity in describing, developing, and analyzing. This applies to all scientific and engineering disciplines as well and in particular for the development of software and data-intensive systems: From the beginning, models have been used here as instruments for (requirements) specification and documentation. Approaches like *Model-Driven Software Development - MDSD*, *Model-Driven Architecture - MDA*, and *Model as a Program – MaaP* produce software out of models, supported by metamodeling frameworks, transformers, generators, "programming machines" etc.

In 2017, the *Model-Centered Architecture – MCA* paradigm (Mayr et al., 2017) was first introduced. According to this paradigm, all processes in a digital system and all data they process are instances of models. These models in turn are instances of meta-models, described using an appropriate modeling language, and represented using a corresponding representation language. Consequently, all system interfaces are defined through models as well. In this way, any digital system comes as a construct of co-operating model handlers (model consumers and/or producers). Together with the handled models it thus can be seen as a *digital twin* or *digital shadow* of the real-world part of the ecosystem to which it is coupled. For modeling general purpose languages such as the Unified Modeling Language UML, domain-specific languages or, of course, combinations can be used.

This Frontiers Research Topic highlights recent work in the area of model-centered systems development. In our Call for Papers we have focused on conceptual modeling as an instrument for the realization of systems. For, conceptual models are particularly suitable for this purpose, as they combine the three key dimensions: *conceptualizations* (modeling concepts, usually defined by meta-model hierarchies), *representations* (linguistic, syntactic models), and the *semantic foundation* of the modeling concepts, for example by means of ontologies (Mayr and Thalheim, 2021).

Fortunately, we received quite a number of interesting papers, of which the five now published papers were accepted in a multi-stage review process. They offer a broad spectrum of approaches, solutions, and insights to model-centric system development.

Jonsson, for example, in his paper "*Conceptual data systems architecture principles for information systems*" proposes as an architectural basis a separation of the conceptual domain of user communities from the information technology domain of a system and models the user domain in three layers: conceptual data logic model, interface model, and user community model. The technical domain is a platform that enables modeling and execution of such a model. As an advantage of this separation, Tomas mentions a "pure conceptual space" of users, in which developers and users can focus on the same concepts and "speak the same language." A participative and iterative development process then builds on this.

Kohan et al. address the area of IoT systems in their paper "*A survey on the model-centered approaches to conceptual modeling of IoT systems*" and present a "mini-survey" on the state of model-centered approaches in this area. For this purpose, they evaluated the following five academic publication repositories for the period from January 2010 to July 2022: SCOPUS, Science Direct, ACM Digital Library, IEEE Explore and SpringerLink. From an initial recall of 952 publications 148 were finally included in the analysis in a multi-stage selection process. Most of these papers introduce a specific conceptual model or a new modeling language or method and the like. Architectural design or fundamental discussions, on the other hand, are the subject of only <25% of the papers. The authors therefore conclude that while there is a large body of research on conceptual modeling of IoT systems, there is a lack of generally accepted approaches and formal methodologies. In particular, the high degree of heterogeneity in IoT technology is a hurdle for holistic model-based analysis.

Prinz et al. take a more fundamental approach to our topic in their paper "*Models, systems, and descriptions - A cross-disciplinary reflection on models*" by considering the differences between physical and mental models and between static and dynamic models. As a framework for meaning-making, they draw on semiotics to identify commonalities between models in different domains. In doing so, they distinguish systems, models, descriptions of systems and descriptions of models to better understand the commonalities between mental and physical models in different domains.

Complex multi-domain systems pose particular challenges for modeling and realization, since they usually involve models formulated in different modeling languages, and therefore need to be harmonized. This is addressed by Latifaj et al. in their paper "*Higher-order transformations for the generation of synchronization infrastructures in blended modeling.*" They propose an automated solution for generating synchronization transformations in an industrial setting. This approach is essentially based on the specification of mapping rules between two domain-specific modeling languages and of the automatic generation of synchronization model transformations based on these rules. A "mapping model language" is proposed to formulate the rules. Technically, a solution for modeling environments is provided based on the Eclipse Modeling Framework (EMF) and DSMLs described using EMF's meta-metamodel, Ecore.

Also dedicated to transformation is the paper "*Preserving conceptual model semantics in the forward engineering of relational schemas*" by Guidoni et al. However, the framework here is much more specific, as it deals exclusively with the generation of relational schemas from conceptual models (and this without semantic loss, if possible, compared to conventional approaches). The approach is based on OntoUML and comes with a tool implementation as proof of concept.

Of course, this collection of papers is far from the complete coverage of the field of model-centered system development. Therefore, we see our Research Topic rather as an impulse for further research and development in this area. We would like to thank the people responsible at Frontiers for making this impulse possible.

Author contributions

HM: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. BT: Methodology, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Mayr, H. C., Michael, J., Ranasinghe, S., Shekhovtsov, V. A., and Steinberger, C. (2017). "Model centered architecture," in *Conceptual Modeling Perspectives*, ed J. Cabot (Cham: Springer), 85–104.

Mayr, H. C., and Thalheim, B. (2021). The triptych of conceptual modeling: a framework for a better understanding of conceptual modeling. *Software Syst. Model.* 20, 7–24. doi: 10.1007/s10270-020-00836-z