

Envisioning the Future of Collaborative Model-Driven Software Engineering

Davide Di Ruscio, Mirco Franzago, Henry Muccini
DISIM Department
University of L'Aquila, Italy
{firstname.lastname}@univaq.it

Ivano Malavolta
Department of Computer Science
Vrije Universiteit Amsterdam, The Netherlands
i.malavolta@vu.nl

Abstract—The adoption of Model-driven Software Engineering (MDSE) to develop complex software systems in application domains like automotive and aerospace is being supported by the maturation of model-driven platforms and tools. However, empirical studies show that a wider adoption of MDSE technologies is still an issue. One limiting factor is related to the limited support for *collaborative MDSE*. This paper reflects on research directions, challenges, and opportunities of collaborative MDSE.

Keywords—Model-Driven Engineering; Collaborative Software Engineering; Collaborative MDSE;

I. INTRODUCTION

Model-driven software engineering (MDSE) [1, 2] is one of the proposed approaches aiming at taming the complexity of software systems via abstraction [3]. In MDSE, models are considered as first-class entities, which are employed both for descriptive and prescriptive purposes. Many are the success stories of MDSE, ranging from the development of large-scale enterprise Web applications, to clinical data management, and to public authority data interchange [2, 4]. On a different line of research, the prominence of agile methods, open-source software projects, and global software development techniques allows software development teams (but also external and non-technical stakeholders) to seamlessly collaborate to engineer complex systems. Such techniques belong to the domain of *collaborative software engineering* (CoSE) [5]. The focus of this paper is on **collaborative model-driven software engineering**, i.e., the intersection between MDSE and CoSE. Collaborative MDSE is gaining a growing interest in both academia and industry [6, 7]. A number of research initiatives are being run to enable large teams of modelers to work on (large) models in a collaborative manner [6], each of them proposing different building blocks and perspectives about collaborative MDSE.

In this paper we explore the main constituent elements of collaborative MDSE, they are synthesized from (i) a systematic map on collaborative MDSE [8, 9], (ii) a thorough discussion occurred during a dedicated workshop called COMMitMDE, and (iii) the interaction with industrial partners. Then, we elaborate on visions of the *future* of collaborative MDSE.

II. COLLABORATIVE MDSE

MDSE provides suitable techniques and tools for specifying, manipulating, and analyzing software modeling ar-

tifacts including metamodels, models, and transformations. Collaborative MDSE focuses on those approaches in which several distributed technical and/or non-technical stakeholders collaborate to produce models of a software system, working in a local or remote shared workspace, either synchronously or asynchronously. Collaborating stakeholders can include, but are not limited to, developers, domain experts, managers, customers, and users of software systems.

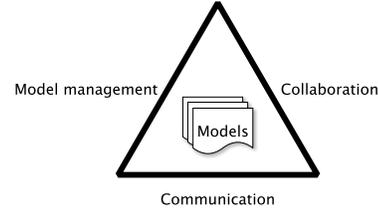


Fig. 1. Collaborative MDSE dimensions

Collaborative MDSE is gaining growing interests in both academia and practice, and during the last few years several studies scattered across different independent research areas have been exploring the “building blocks”, which are required to support collaborative MDSE. Modeling editors, model repositories, model versioning systems, and model differencing are only few examples of techniques and tools required when employing collaborative MDSE approaches. Starting from this body of knowledge, it is possible to identify the following three complementary dimensions, which can be considered as the peculiar aspects building up a collaborative MDSE approach (see Figure 1) [9]:

- ▷ *model management infrastructures* for managing the life cycle of models (e.g., modeling tools, modeling repositories);
- ▷ *collaboration means* for allowing involved stakeholders to work on the modelling artifacts collaboratively (e.g., model versioning systems, model merging mechanisms, systems for visualizing and managing model conflicts);
- ▷ *communication means* for allowing involved stakeholders to be aware of what the other stakeholders collaborating with them are doing (e.g., chat, social network, wiki, asynchronous messaging system, issue tracker).

III. VISIONS OF THE FUTURE

In the following we detail the envisioned aspects with respect to the three collaborative MDSE dimensions, concluding with (one of) the main orthogonal challenges.

Model Management. Modeling one single small model by means of a stand-alone tool is surely a thing from the past. In the future, we envision a *distributed modeling environment* in which *multiple stakeholders* of different types (e.g., developers, end users, sales) edit (part of) models, possibly in *real-time*. This scenario opens up for many challenges, such as the support of *light-weight, Web-based modeling environments* (it is unthinkable to ask a team member to install a modeling environment locally today), accessible by following a *multi-device experience*, which facilitates collaboration in *mobility*.

High-volumes of data are generated by everything around us everyday; models of this data (potentially generated at runtime) will follow the same trend in the very recent future. Even if there are some approaches advocating the capability of managing large scale models [10], we will need more advanced techniques to store, maintain, and edit models in a *scalable* manner, and *cloud-based services* for manipulating or dynamically extracting parts of them [11, 12].

Initial steps in these directions include: partial loading of models [13], model pre-fetching [14], incremental model transformations [15], browser-based model editors with cloud services [16]. A point of inspiration in a different domain is the Eclipse Che project (<http://www.eclipse.org/che>), it is an on-demand, web-based collaborative IDE where tools, code dependencies, and workspaces can be dynamically provisioned.

Collaboration. More than ten years ago, we all have been amazed by the real-time collaborative features of Google Docs. We envision a similar revolution in MDSE: stakeholders will collaborate either in *real-time* or in an *asynchronous* way on shared models belonging to shared *ecosystems of modeling artifacts*. Models in those ecosystems will be *queried*, edited, *mapped* to each other, *transformed in a light-weight manner*, or even injected into the running system (e.g., via *model interpretation or compilation*). In this scenario it will be fundamental to have *inconsistency-tolerant environments* for making collaboration smooth, *model repositories* with extraordinary capacity and performance, advanced engines for *detecting and resolving inconsistencies*, *branching* and *merging* capabilities, possibly with as much automation as possible.

Interesting initial steps in this directions include: inconsistency management via process transformation, access control rules at the model element level [17], incremental bidirectional model transformations [15], model indexing and efficient querying [18], NoSQL databases for model repositories with nearly instant read/write time [19]. Also, when talking about collaborative modeling, we are implicitly bringing up the concept of *multi-view modeling*, where multiple stakeholders may be working on multiple views of the same system at the same time. In turn, multi-view modeling brings up the well-known tension between the *projective* and the *synthetic* multi-view principles. If on one side projective approaches may simplify collaboration (mainly because the common metamodel is the only source or target for the enactment of editing operations), synthetic approaches may introduce accidental complexity (mainly due to the potentially large number of interdependences among modeling artifacts [20]).

In any case, each multi-view principle may be embraced for many other reasons, either organizational or technical, thus additional scientific effort must be devoted to both of them, even by suitably combining them. As an example, first results in this direction have been proposed with the cloud-based multi-view modeling environment based on AToMPPM [21].

Communication. Software modelling will be performed by teams with members with *very different technical knowledge and background*. Without proper communication means, this situation may lead to mismatches in the used language, understanding of the system, additional design and development iterations, thus resulting in unforeseen waste of resources, budget and time. We see a big research gap in terms of *support for collaboration between technical and non-technical stakeholders*. This trend will be more evident in domains where soft skills are more needed; for example, stakeholders typically involved in the development of mobile applications include user experience designers, graphic designers, information architects, developers, users, and customers [22, 23].

Many collaborative MDSE approaches are performing well in terms of *workspace awareness*, however there is still room for improvement, especially for synchronous approaches, where modeling elements cannot simply disappear or move within the editing environment. In the future we will see advanced facilities for workspace awareness in terms of *who* is performing editing operations (e.g., role-based editing) on *what* model or modeling element (e.g. real-time model updates), and *why* the change has been performed (e.g., quick recording of design decisions and their rationale). As an example, WebGME [24, 25] supports well the *what* dimension via tool status notifications and real-time updates in the modelling editor, but those updates just appear in the editor without any indication about who made them on which parts of the models.

Finally, it will be fundamental to ensure *traceability* between the design decisions discussed in communication-oriented contents (e.g., the text of a chat, the page of a Wiki) and modeling artifacts (e.g., a model, a specific model element, or even previous versions of a model). Indeed, those (potentially typed or structured) links will be a precious asset for reusing technical and non-technical knowledge about the models developed within large software organizations.

Main Orthogonal Challenge. During the previously described analysis and reasoning, we came across a common future challenge, which is orthogonal to all the dimensions: *flexibility*. It is difficult to imagine a totally generic collaborative MDSE approach that perfectly fits any application domain (e.g., automotive vs. Web apps), any organization (large corporates vs. startup-like companies), any development process (certification-oriented development vs. SCRUM-based agile development) [26]. Modular and plugin-based architectures will play a role when architecting a collaborative MDSE tool. We envision flexible and agile modeling infrastructures where any stakeholder can add, customize, or remove the functionalities of the modeling environment at wish, thus adapting it to the (ever-changing) specific project and organizational needs.

REFERENCES

- [1] J. Bézivin, “On the unification power of models,” *Software & Systems Modeling*, vol. 4, no. 2, pp. 171–188, 2005.
- [2] J. Hutchinson, M. Rouncefield, and J. Whittle, “Model-driven engineering practices in industry,” in *Proceedings of the 33rd International Conference on Software Engineering*. ACM, 2011, pp. 633–642.
- [3] J. Kramer, “Is abstraction the key to computing?” *Communications of the ACM*, vol. 50, no. 4, pp. 36–42, 2007.
- [4] D. Di Ruscio, R. F. Paige, and A. Pierantonio, “Guest editorial to the special issue on success stories in model driven engineering,” *Science of Computer Programming*, vol. 89, no. PB, pp. 69–70, 2014.
- [5] I. Mistrík, J. Grundy, A. Van der Hoek, and J. Whitehead, “Collaborative software engineering: challenges and prospects,” in *Collaborative Software Engineering*. Springer, 2010, pp. 389–403.
- [6] D. S. Kolovos, L. M. Rose, N. Matragkas, R. F. Paige, E. Guerra, J. S. Cuadrado, J. De Lara, I. Ráth, D. Varró, M. Tisi *et al.*, “A research roadmap towards achieving scalability in model driven engineering,” in *Proceedings of the Workshop on Scalability in Model Driven Engineering*. ACM, 2013, p. 2.
- [7] M. Brambilla, J. Cabot, and M. Wimmer, *Model-driven software engineering in practice*. Morgan & Claypool Publishers, 2012, vol. 1, no. 1.
- [8] M. Franzago, D. Di Ruscio, I. Malavolta, and H. Muccini, “Protocol for a Systematic Mapping Study on Collaborative Model-Driven Software Engineering,” DISIM - University of L’Aquila, Tech. Rep. TR-001-2016, <https://arxiv.org/pdf/1611.02619.pdf>.
- [9] D. Di Ruscio and M. Franzago, “A classification of collaborative model-driven software engineering approaches: Opportunities and challenges,” in *Proceedings of the 1st International Workshop on Collaborative Modelling in MDE (COMMtMDE) co-located with MODELS conference*, 2016.
- [10] D. Kolovos, L. Rose, R. Paige, E. Guerra, J. Cuadrado, J. De Lara, I. Ráth, D. Varró, G. Sunyé, and M. Tisi, “Mondo: scalable modelling and model management on the cloud,” in *STAF2015 Project Showcase*, 2015.
- [11] F. Basciani, J. Di Rocco, D. Di Ruscio, A. Di Salle, L. Iovino, and A. Pierantonio, “Mdeforge: an extensible web-based modeling platform,” in *CloudMDE@ MoDELS*, 2014, pp. 66–75.
- [12] J. Di Rocco, D. Di Ruscio, L. Iovino, and A. Pierantonio, “Collaborative repositories in model-driven engineering,” *IEEE Software*, vol. 32, no. 3, pp. 28–34, May 2015.
- [13] R. Wei, D. S. Kolovos, A. Garcia-Dominguez, K. Bampis, and R. F. Paige, “Partial loading of xmi models,” in *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*. ACM, 2016, pp. 329–339.
- [14] G. Daniel, G. Sunyé, and J. Cabot, “Prefetchml: a framework for prefetching and caching models,” in *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*. ACM, 2016, pp. 318–328.
- [15] O. Semeráth, A. Vörös, and D. Varró, “Iterative and incremental model generation by logic solvers,” in *International Conference on Fundamental Approaches to Software Engineering*. Springer, 2016, pp. 87–103.
- [16] J. Gray and B. Rumpe, “The evolution of model editors: browser- and cloud-based solutions,” *Software & Systems Modeling*, vol. 15, no. 2, pp. 303–305, 2016.
- [17] G. Bergmann, C. Debreceni, I. Ráth, and D. Varró, “Query-based access control for secure collaborative modeling using bidirectional transformations,” in *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*. ACM, 2016, pp. 351–361.
- [18] G. Daniel, G. Sunyé, and J. Cabot, “Mogwai: a framework to handle complex queries on large models,” in *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*. IEEE, 2016, pp. 1–12.
- [19] A. Benelallam, A. Gómez, G. Sunyé, M. Tisi, and D. Launay, “Neo4emf, a scalable persistence layer for emf models,” in *European Conference on Modelling Foundations and Applications*. Springer, 2014, pp. 230–241.
- [20] D. D. Ruscio, I. Malavolta, H. Muccini, P. Pelliccione, and A. Pierantonio, “Model-Driven Techniques to Enhance Architectural Languages Interoperability,” in *Fundamental Approaches to Software Engineering (FASE)*. Springer Berlin/Heidelberg, 2012, pp. 26–42.
- [21] J. Corley, E. Syriani, H. Ergin, and S. Van Mierlo, “Cloud-based multi-view modeling environments,” in *Modern Software Engineering Methodologies for Mobile and Cloud Environments*. IGI Global, 2016, pp. 120–139.
- [22] M. Franzago, H. Muccini, and I. Malavolta, “Towards a collaborative framework for the design and development of data-intensive mobile applications,” in *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems*. ACM, 2014, pp. 58–61.
- [23] M. Franzago, I. Malavolta, and H. Muccini, “Stakeholders, viewpoints and languages of a modelling framework for the design and development of data-intensive mobile apps,” *CoRR*, vol. abs/1502.04014, 2015. [Online]. Available: <http://arxiv.org/abs/1502.04014>
- [24] M. Maróti, T. Kecskés, R. Kereskényi, B. Broll, P. Völgyesi, L. Jurácz, T. Levendovszky, and Á. Lédeczi, “Next generation (meta) modeling: Web- and cloud-based collaborative tool infrastructure,” in *MPM@ MoDELS*, 2014, pp. 41–60.
- [25] M. Maróti, R. Kereskényi, T. Kecskés, P. Völgyesi, and Á. Lédeczi, “Online collaborative environment for designing complex computational systems,” *Procedia Computer Science*, vol. 29, pp. 2432–2441, 2014.
- [26] D. Di Ruscio, A. Pierantonio, and J. de Lara, “Extreme Modelling (XM) 2012 Special Section,” *Journal of Object Technology*, vol. 13, no. 3, Jul. 2014, (editorial). [Online]. Available: http://www.jot.fm/contents/issue_2014_07/xm2012.html