The Journal of Systems & Software 199 (2023) 111626

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Collaborative Model-Driven Software Engineering — A systematic survey of practices and needs in industry[☆]

Istvan David[a,*], Kousar Aslam[b], Ivano Malavolta[b], Patricia Lago[b]

[a] Department of Computer Science and Operations Research, Université de Montréal, Canada
[b] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 5 September 2022
Received in revised form 1 November 2022
Accepted 24 January 2023
Available online 26 January 2023

Keywords:
Model-driven engineering
Collaborative software engineering
Industry survey

ABSTRACT

The engineering of modern software-intensive systems is carried out in collaboration among stakeholders with specialized expertise. The complexity of such systems often also necessitates employing more rigorous approaches, such as Model-Driven Software Engineering (MDSE). Collaborative MDSE is the combination of the two disciplines, with its specific opportunities and challenges. The rapid expansion and maturation of the field started attracting tool builders from outside of academia. However, available systematic studies on collaborative MDSE focus exclusively on mapping academic research and fail to identify how academic research aligns with industry practices and needs. To address this shortcoming, we have carried out a mixed-method survey on the practices and needs concerning collaborative MDSE. First, we carried out a qualitative survey in two focus group sessions, interviewing seven industry experts. Second, based on the results of the interviews, we constructed a questionnaire and carried out a questionnaire survey with 41 industry expert participants. In this paper, we report the results of our study, investigate the alignment of academic research with the needs of practitioners, and suggest directions on research and development of the supporting techniques of collaborative MDSE.

1. Introduction

The intricacies of nowadays' software engineering processes require a coordinated interplay between stakeholders and engineers of different expertise, giving rise to large and highly heterogeneous teams. These teams are typically distributed in space (e.g., different workspaces or countries), and often in time as well (e.g., different work shifts or time zones). *Collaborative software engineering* (Herbsleb, 2007; Whitehead, 2007) aims to support such teams in their efficient teamwork. The complexity of modern software-intensive systems requires rigorous formal engineering methods. Pertinent examples include mechatronic and cyber–physical systems (Persson et al., 2013). *Model-driven software engineering (MDSE)* answers these needs by addressing the inability of programming languages to alleviate the complexity of platforms and express domain concepts effectively (Schmidt, 2006). By that, MDSE provides stakeholders and engineers with techniques for reasoning about the system at levels of abstraction higher than that of source code, allowing for abstract rigorous techniques, such as the validation and verification of the system, and highly automated software construction by code generation.

As the combination of collaborative software engineering and MDSE, *collaborative MDSE* exhibits the traits of both worlds and presents its specific benefits and challenges (Muccini et al., 2018). Collaborative MDSE has become a prominent feature of nowadays' software engineering practice (Brambilla et al., 2017), e.g., in agile methodologies and low-code platforms (Zhang and Patel, 2011; Sanchis et al., 2019; Cabot, 2020). The rapid expansion and maturation of the field started attracting tool builders from outside of academia as well.[1][2][3] Academic research still has to pave the way by developing novel methods and techniques for the future generation of collaborative MDSE tools. The available systematic studies on collaborative MDSE, however, focus exclusively on mapping and classifying academic research (Franzago et al., 2018; David et al., 2021a). Mapping the state-of-the-practice, its shortcomings, preferences, and needs have been lacking so far, leaving academic researchers without a firm lead to steer their work.

In this paper, we address this shortcoming by reporting on our survey of the industry practices and needs related to collaborative MDSE.

[☆] Editor: Matthias Galster.
[*] Corresponding author.
E-mail address: istvan.david@umontreal.ca (I. David).

[1] http://www.mondo-project.org
[2] https://itea4.org/project/bumble.html
[3] https://www.lowcomote.eu

https://doi.org/10.1016/j.jss.2023.111626
0164-1212/

We have carried out our study in two steps. First, we have organized two focus group discussions between February–March 2021 with five industry experts to evaluate and validate our initial hypotheses on the practices and needs of the industry, based on (i) the industry requirements elicited in one of our large-scale projects[4]; and (ii) on the known systematic studies on the academic research on collaborative MDSE (David et al., 2021a; Franzago et al., 2018; Masson et al., 2017). Second, based on the takeaways from the focus group discussions, we have constructed a questionnaire of specific features of collaborative MDSE, and asked a wider audience of practitioners between June–July 2021 and February–March 2022 to evaluate every feature in terms of (i) current frequency of usage; and (ii) the utility and need for the particular feature. We were interested in the subset of practitioners who are *users* of collaborative MDSE in the first place, not tool providers for its support. Our study is inclusive to all models that fit the definition of MDSE (Schmidt, 2006; Brambilla et al., 2017), irrespective of their application domain, technology, and role in the overall product lifecycle. Eventually, we have recorded the input of 41 industry experts. To enable the validation and reproduction of our study, we have published all data (including transcripts of focus group discussions, the questionnaire, the extracted data, and the analysis scripts) in a replication package.[5]

The main **contributions** of this study are the following:

- a classification framework for mapping the practices and needs of industry in collaborative MDSE;
- identification of current practices and needs;
- elicitation of insights relevant to the target audience;
- the complete replication package of the study.

The **target audience** of this study is composed of (i) academic researchers and (ii) tool providers. Researchers can use our findings for getting an overview of the needs of industry and steer their research towards high-impact and industry-relevant topics within collaborative MDSE. Tool providers (and technology transfer entities) can use our findings to identify currently needed collaborative MDSE features, anticipate the capabilities of the next generation of collaborative MDSE tools, and prepare for the associated challenges. Additionally, practitioners can use the findings of this study to understand the state of the practice, their position within it, and identify adoptable, mature techniques for collaborative modeling.

The rest of this paper is structured as follows. In Section 2, we provide a brief overview of the background of our work, including the classification framework used throughout this study. In Section 3, we elaborate on the design and execution of the study. In Section 4, we present the results, and discuss them in Section 5. In Section 6, we discuss the threats to validity. We conclude the paper by reviewing the related work in Section 7, and drawing the conclusions in Section 8.

2. Collaborative MDSE

Collaborative MDSE is the application of principles of collaborative software engineering to the general domain of MDSE. Relying on formal models as the primary artifacts of the engineering process enables the early analysis and verification of the system to be built (Bang et al., 2017). With its increasing adoption in practical settings (Bucchiarone et al., 2021), scaling MDSE to large, hierarchical teams has become a pressing need (Kolovos et al., 2013), necessitating the development of collaborative means for MDSE.

Models are typically serialized in files, governed by an appropriate physical metamodel, such as XMI for the Meta-Object Facility (MOF).[6] This enables employing collaborative MDSE techniques of traditional software engineering to support collaborative MDSE. To fully leverage the additional semantic information encoded in models, more sophisticated techniques have been introduced, often operating at higher levels of abstraction. Examples include domain-specific (Zadahmad et al., 2019) and semantic (Kehrer et al., 2011) model differencing, model-based difference visualization (Brosch et al., 2012), semantic inconsistency management (David et al., 2016a), and intelligent editing assistance (Mussbacher et al., 2020). We aim to map which of these techniques are frequently encountered; and which ones are sought after by practitioners.

According to our previous studies (Franzago et al., 2018; David et al., 2021a) on the academic literature, collaborative MDSE approaches can be organized according to three orthogonal dimensions, namely: model management, collaboration, and communication. *Model management* defines techniques and tools for managing the lifecycle of models, including their creation, manipulation, and storage. *Collaboration* defines techniques and tools that enable effective and efficient groupwork across the involved stakeholders. Typical means of collaboration in MDSE include versioning systems with merging and branching support, consistency management mechanisms, and conflict resolution algorithms. *Communication* features allow a semantically rich exchange among the involved stakeholders, to augment the information carried by the models they collaborate over. Typical means of communication are chats, wikis, model annotations, comments, and change proposals, many of which are present in modern issue tracking systems like Jira.

3. Study design and execution

This study is carried out according to well-established guidelines for empirical software engineering (Wohlin et al., 2012), focus groups (Kontio et al., 2008), and survey design (Kitchenham and Pfleeger, 2002). In the remainder of this section, we present (i) a formulation of the goal and research questions of the study (Section 3.1) and (ii) an overview of the study design and execution (Section 3.2).

3.1. Goal and research questions

The goal of this study is to identify, organize and analyze the current practices and needs in collaborative MDSE from an industry practitioner's point of view. We formulate the following four research questions to guide our study.

RQ1. *What is the level of adoption of Collaborative MDSE features by practitioners?*

By answering this question, we aim to identify the main trends in the state of the practice. We assume that the frequency of usage is an appropriate indicator of adoption; and that a technique or solution is frequently used because (i) industry processes are capable to accommodate them, and (i) the required tool support is available. Researchers can use this information for validating their assumptions regarding the adoption of collaborative modeling features in practice. Tool builders can use this information for identifying sought-after features with a high market share. Industry entities can use this information for identifying mature techniques with a likely reduced risk.

4 https://itea4.org/project/bumble.html

5 https://doi.org/10.5281/zenodo.7588593

6 https://www.omg.org/mof/

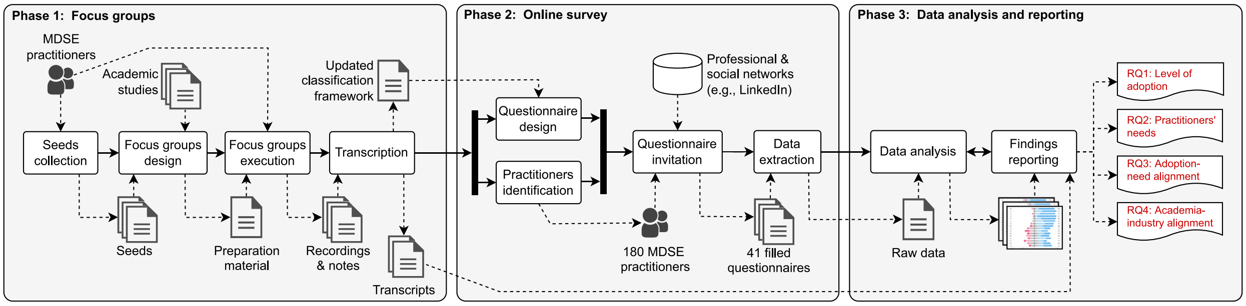


Fig. 1. Overview of the study.

RQ2. *What are the practitioners' needs with respect to Collaborative MDSE features?*

By answering this question, we aim to identify Collaborative MDSE features with high added value. We assume that a need in practice is present either because of (i) the lack of topical basic research, or (ii) the lack of appropriate tooling transferring the results of topical basic research to practitioners. Researchers tool builders can use this information to identify research topics which, in addition to academic basic research, might require further industry evangelization before getting adopted.

RQ3. *How does the adoption of, and the need for Collaborative MDSE features relate?*

By answering this question, we aim to understand whether the needs of practitioners indeed align with the adoption of Collaborative MDSE features. The mismatch between need and adoption might indicate features that are sought-after but not supported currently. Researchers can use this information to steer their research towards applicable directions. The alignment of need and adoption might indicate properly supported Collaborative MDSE features. Tool builders can use this information to support their decisions when aiming to choose mature techniques for their prospective tools. Finally, industry decision makers can use these information to reduce technological risks and to gain a competitive advantage by the early adoption of emerging solutions.

RQ4 *How do the current trends of academic research align with the needs of practitioners w.r.t. Collaborative MDSE?*

By answering this question, we primarily aim to identify any topics that are needed by the practice but are underrepresented in the state of the art. By that, we aim to aid researchers and industry entities in establishing new, possibly joint strands of research addressing the foundations of sought-after collaborative modeling techniques and solutions; and to aid the broader academic community in establishing the proper overall roadmap for the next generation of topical research, including research projects, scientific venues, and industry outreach.

We remark that eventually, 41 participants were sampled in our study, and the research questions were answered based on their input. This number limits the generalizability of results, as explained in Section 6.

3.2. Overview of the study design and execution

Our study follows a *mixed-method research method*. As shown in Fig. 1, our study is designed as a three-phase process. In Phase 1 we conducted two **focus group sessions** in order to validate and improve the classification framework (Table 2). We decided to adopt the focus group method since it is an efficient empirical approach for obtaining rich qualitative insights and feedback from practitioners (Merriam and Tisdell, 2015), and has been successfully adopted in software engineering research (Kontio et al., 2004). In Phase 2 we carried out an **online survey** using an online questionnaire constructed from the updated classification framework resulting from phase 1. We opted for this method to reach as many practitioners as possible, taking into consideration their geographical distribution and heterogeneity in terms of the types of projects and companies they work in. Finally, in Phase 3 we **analyzed and reported** the data collected from the online survey and the focus groups to answer our research questions.

In the following, we review the three phases of our study. Further details are available in the replication package.[1]

3.2.1. Phase 1: Focus groups

This phase has the following main goals: (i) to familiarize ourselves with collaborative MDSE in industry settings, (ii) to identify relevant areas of interest with respect to collaborative MDSE in industry, (iii) to collect qualitative data about collaborative MDSE in industry, which will be used in phase 2 for designing the questionnaire and in phase 3 to complement the quantitative data coming from the online survey. As shown in Fig. 1, this phase is organized into four main steps: seeds collection, focus groups design, focus groups execution, and transcription.

Seeds collection. In this step, we contacted MDSE practitioners involved in projects where collaborative MDSE is part of the development activities. To ensure that we were targeting the right subjects, we decided to select them by convenience sampling (Wohlin et al., 2012). Specifically, we directly contacted all industry partners within the consortium of the BUMBLE ITEA3 project[7], a European research project centered on blended modeling (David et al., 2022a; Latifaj et al., 2021) and collaborative modeling (Franzago et al., 2018). We asked those practitioners to provide us a list of *seeds* about collaborative MDSE; to not restrict or bias too much the input of the practitioners, in the invitation we defined the seeds broadly, as *requirements, needs, challenges, really anything related to collaborative modeling in your company*. We collected a total of 44 individual seeds, provided by 8 different companies across The Netherlands, Sweden, and Austria. The involved companies are Canon Printing Technologies, HCL, Volvo, Eclipse Source, AVL, Unibap, Sioux Technologies, and Modeling Value Group.

Every seed is composed of the following parts: unique ID, description, involved stakeholders, and priority. Due to confidentiality restrictions, we cannot report the complete list of the collected seeds. Representative examples of collected seeds include the following.

- *"Model-based diff and merge with Git integration using textual notations."*
- *"Cope with different model editing rights depending on view and expertise."*

7 https://itea4.org/project/bumble.html

Table 1
Focus group participant demographics.

FG	ID	Role	Ex	Domain	CS
FG1	P1	Team lead	20	Embedded SW	L
FG1	P2	Tech. scout for MDSE	14	Automotive	XL
FG1	P3	Software dev. manager	18	Consulting	XXL
FG2	P4	Model-based developer	8	Printing	L
FG2	P5	Product owner	36	Consulting	XS

FG: focus group ID; ID: Participant identifier; Ex: years of industry experience; Domain: app. domain of the company; CS: company size (XS<20; S<100; M<500; L<5K; XL<10K; XXL>10K).

- *"Real-time (synchronous) collaborative modeling. Automated conflict resolution."*

The collected seeds were manually analyzed by two researchers and iteratively organized into emerging clusters, such as model versioning, process & integration, support for multi-notation, etc. They formed the basis of the points to be discussed during the focus group sessions.

Focus groups design. In this step, we designed the focus group by following the guidelines by Kontio et al. (2008). The goal of the focus group sessions is twofold: (i) to consolidate our knowledge on collaborative MDSE in the industry (which in turn will be used for designing the online questionnaire in phase 2); (ii) to collect qualitative data on how MDSE practitioners perceive and talk about collaborative MDSE (this data will complement the quantitative data collected from the online survey).

In line with methodological guidelines (Kontio et al., 2008), the sessions lasted 90 min and were organized into four phases.

1. *Introduction* (5 min). A round of introduction among the participants and the moderator that fosters group dynamics, allows participants to gain confidence to speak freely and to gain information about each other.
2. *Overview of the topic* (5 min). The moderator describes the overall goal of the research (see Section 3.1), its link with the focus group session, and discusses the background on collaborative MDSE to set a general common ground on the topic guiding the ensuing discussion.
3. *Discussion* (70 min). This phase is organized by the three dimensions of collaborative MDSE (see Section 2), enriched with (i) additional systematic *Academic studies* on collaborative MDSE (Franzago et al., 2018; David et al., 2021a; Masson et al., 2017) and (ii) the seeds elicited in the previous step. For each collaborative MDSE dimension, the moderator gave a conversation starter by hypothesizing a list of MDSE features belonging to the current dimension, and then, the group discussed this hypothesis. The conversation among participants was guided by the moderator by asking participants to elaborate on the following questions. (i) Is the hypothesis in line with your experience? (ii) Is there anything missing, according to your experience? (iii) Is there anything new or unexpected to you, according to your experience?
4. *Wrap-up* (10 min). After the discussion, the participants were given the option to reflect on the session or comment in an open-ended way.

The participants were informed about the goals and tentative structure of the session in document which also described the three dimensions of collaborative MDSE.

Focus groups execution. We invited all practitioners involved in the BUMBLE project (see the previous step), leading to five participants overall. Table 1 reports the demographics of the participants of both focus group sessions.

Based on the availability of the participants, we organized two focus group sessions between February and March 2021. Three researchers were present in all sessions and one of them acted as moderator. The two researchers not moderating the session took notes about the main points of the discussion. Both focus group sessions were conducted virtually and were video recorded for further analysis.

Transcription. Two researchers manually transcribed the video recordings of the focus group sessions by following the denaturalism approach (Oliver et al., 2005), i.e., the grammar was corrected, the interview noise was removed, and non-standard accents were standardized while ensuring a full and faithful transcription. The full contents of the transcriptions are available in the replication package of this study (see Section 3.3).

Three researchers subsequently analyzed the transcripts to update the classification framework for collaborative MDSE. We recall that the original version of the classification framework was based on our previous peer-reviewed systematic mapping studies on collaborative MDSE (Franzago et al., 2018; David et al., 2021a) and the feature model of collaborative modeling defined in Masson et al. (2017). The resulting classification framework is shown in Table 2.

In addition to low-level updates of various features in the framework, the most relevant improvement entails the addition of intermediate feature groups within each dimension. Specifically, the model management dimension has been further detailed into three feature groups: *Models and languages*, *Model manipulation*, and *Editors and modeling environments*. The five new feature groups of the collaboration dimension are *Stakeholder management and access control*, *Collaboration dynamics*, *Versioning*, *Conflicts and consistency*, *Network architecture and robustness*. Finally, the three new feature groups of the communication dimension are *Synchronicity*, *Asynchronicity*, and *Integration*. The main motivation for having the intermediate feature groups is to make the classification framework more cognitively manageable and to better characterize the collaborative MDSE domain. A complete description of the features and the new feature groups of the classification framework are provided in the remainder of this paper, in the replication package (see Section 3.3), and in Franzago et al. (2018), David et al. (2021a), Masson et al. (2017). We use this updated framework to design the online questionnaire (Section 3.2.2) and to report the results (Section 4).

3.2.2. Phase 2: Online survey

The goal of this phase is to collect practices and needs from as many MDSE practitioners as possible involved in collaborative MDSE activities. As shown in Fig. 1, this phase is composed of four main steps: questionnaire design, practitioners identification, questionnaire invitation, and data extraction.

Questionnaire design. In this step we followed well-established guidelines for questionnaire design (Shull et al., 2007; Oppenheim, 2000). The questionnaire is composed of 44 questions. To alleviate the cognitive load on the participants, the majority of the questions are closed-ended (30) and the rest are open-ended (14). The open-ended questions allow participants to freely discuss their individual experiences.

The questionnaire was implemented in Google Forms[8] and is organized into seven sections.

[8] https://forms.google.com

Table 2
Classification framework for collaborative MDSE.

MODEL MANAGEMENT (29)
Models and languages
Collaboration at the model level
Collaboration at the metamodel level
Multi-view modeling (e.g. different views for different stakeholders)
Use of general-purpose modeling languages (e.g., UML, CAD)
Use of domain-specific languages
Import of an external language into the modeling environment
Model manipulation and query
Model validation
Model execution
Model debugging
Model browsing/searching
Model testing by defining the test cases in the models
Lazy loading of the models/workspace
Round-trip engineering (from model to code and back)
Code generation
Model transformation
Integration with build/DevOps tools (e.g., cmake, Jenkins)
Database integration
Metrics of model complexity
Natural Language Processing (for model building)
Editors and modeling environments
Visual editors
Textual editors
Tabular editors
Tree-based editors
Sketch-based editors
Editors supporting multiple types of notations
Projectional editors
Desktop-based modeling environments
Web-based modeling environments
Mobile device based modeling environments
COLLABORATION (27)
Stakeholder management & access control
Role-based access control
Authentication and authorization from corporate database
Anonymous access
User identification
User presence visualization
Collaboration dynamics
Human–Machine collaboration
Real-time collaboration
Offline (non-Real-time) collaboration
Versioning
Model differencing
Model differencing based on the modeling language, not on the file contents
Internal versioning support
External version control (for instance Git, SVN)
Model merging
Version branching
Undo-redo support during collaboration
History
Conflicts and consistency
Locking
Prevention of conflicts
Conflict awareness features (for instance, warnings, prompt actions)
Automation of conflict resolution
Manual conflict resolution
Metrics of degree of conflict/inconsistency
Eventual consistency
Push notifications on conflicts
Network architecture & robustness
P2P (serverless) network architecture
Cloud-based network architecture
Failure recovery
COMMUNICATION (25)
Synchronous communication
Chat
Audio
Voice
Hand gestures
Face-to-face
Change review sessions
Screen sharing
Asynchronous communication
Email
Wiki
Forum
Proposals
Voting
Annotations
Comments
Feedback
Reviews
Call-For-Attention
Sticky notes
Tags
Conflicts table
Multimedia annotations
Commit messages
Integrated professional-social networking
Integration
Communication means built into the modeling tools
Communication means NOT built into the modeling tools

(*continued on next page*)

Table 2 (*continued*).

1. *Introduction*. This section explains the purpose of our study and discloses the privacy and other administrative information. Also, it contains a link to a one-pager[9] for giving the definitions of the terminology that we use about Collaborative MDSE.
2. *Demographics and company information*. Contains questions about general demographics, such as primary background, years of professional experience in MDSE, company information, and their role in it. This data allowed us to analyze the demographics of participants and understand their working environments.
3. *Information about the chosen project (specific to one project)*. To keep the participants focused and collect meaningful information, all questions in the remainder of the questionnaire are given by considering one recent project with collaborative modeling aspects the participant has been involved in. This section collects information about the project, such as its application domain, the average number of collaborators, modeling platform, overall duration, estimated of the size of the software system being developed, and average size of the models. We used this data for the characterization of the projects.
4. *Model management (specific to one project)*. This section focuses on the Model management dimension of the updated collaborative MDSE classification framework (Table 2), and is organized into two sub-sections:
   (a) State of the practice — participants indicate how frequently they have encountered each model management feature in the context of their chosen project.
   (b) Needs — participants assess the usefulness of model management features in future projects.

   We use Likert-type rating scales to guide participants in expressing their answers. Likert-type scales are psychometric rating scales frequently employed in questionnaires to measure the attitude of participants towards a specific question (Joshi et al., 2015). Likert-type scales have been used in practitioner survey in related domains, such as object-oriented software quality (Stevenson and Wood, 2018), search-based refactoring (Simons et al., 2015), and software design patterns (Zhang and Budgen, 2013). Here, we measure the attitude of participants towards statements about the adoption of and

9 Available in the replication package of the study — see Section 3.3.

need for specific collaborative MDSE techniques. The Likert item measuring the adoption of collaborative MDSE techniques is introduced by the following question: "In your project, how frequently did you encounter the following means of model management?" The Likert item defines potential answers on the 5-point scale of {*never, rarely, sometimes, often, always*}. The Likert item measuring the adoption of collaborative MDSE techniques is introduced by the following question: "In a potential future project, how useful would you find the following means of model management?" The Likert item defines potential answers on the 5-point scale of {*definitely not useful, probably not useful, neutral, probably useful, definitely useful*}. We defined two Likert items for each feature in Table 2: one for measuring adoption of the feature; and one for measuring its need. This way, we have obtained $2 \times 81 = 162$ Likert items. At the end of each sub-section, participants were provided with the option of further elaborating in an open-ended comment.
5. *Collaboration (specific to one project)*. This section focuses on the Collaboration dimension of the updated collaborative MDSE classification framework and follows the same structure as the previous one.
6. *Communication (specific to one project)*. This section focuses on the Communication dimension of the updated collaborative MDSE classification framework and follows the same structure as the previous one.
7. *Concluding questions*. This section allows participants to comment on this study in order to identify topics of interest that are important to them but are not mentioned in the questionnaire. Finally, the participants are provided with the options of receiving a preprint of our report and staying in touch for further communication.

Practitioners identification. The target audience of the questionnaire are industry practitioners with experience in collaborative MDSE. Accordingly, it was strictly required that only industry practitioners were recruited for the survey, and we achieved that in two ways.

First, we identified all industry practitioners who published at least a scientific paper at MODELS, the ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems in the past 10 years, both Technical and Industry tracks. MODELS is the flagship scientific conference on MDSE and it has a good history of attracting industry practitioners. We decided to scope our search over the last 10 years to be sure that participants are still active in the field and technologically up-to-date.

Second, we identified all industry practitioners who published at least a scientific paper across all editions of scientific workshops centered on collaborative MDE, specifically: the International Workshop on Collaborative Modelling in MDE (COMMitMDE, 3 editions in 2016, 2017, and 2018) (Bosch et al., 2018) and the International Hands-on Workshop on Collaborative Modeling (HoWCoM, 1 edition in 2021) (David et al., 2021b).

Third, we have compiled the list of practitioners among those belonging to our network of industry collaborators. We contacted them directly, and we selected the set of valid candidates for the online questionnaire among the people who had experience with collaborative MDSE. We extended our set of practitioners by asking prospective participants to also nominate additional experts in their networks (applying the *snowballing sampling approach* (Kitchenham and Pfleeger, 2002)).

Eventually, we identified 180 potential participants.

Questionnaire invitation. In this phase, we reached out to the 180 MDSE practitioners and invited them to participate in the study. To increase the number of participants, we also posted the link to the online questionnaire in thematic groups of professional portals and thematic groups on LinkedIn (such as the MDE network[10] and the Model-Driven Development Forum[11]), and on the social media accounts of one of the authors of this research.

The first round of the survey ran from June 7, 2021, to August 31, 2021, and it was completed by 31 participants. Then, we performed a second round of recruitment between February 9, 2022, and March 1, 2022, which was completed by 10 additional participants. After the two rounds of recruitment, the survey was completed by **41 participants** from at least 38 different companies. (The exact number is unknown due to unidentified participants included by snowball sampling.)

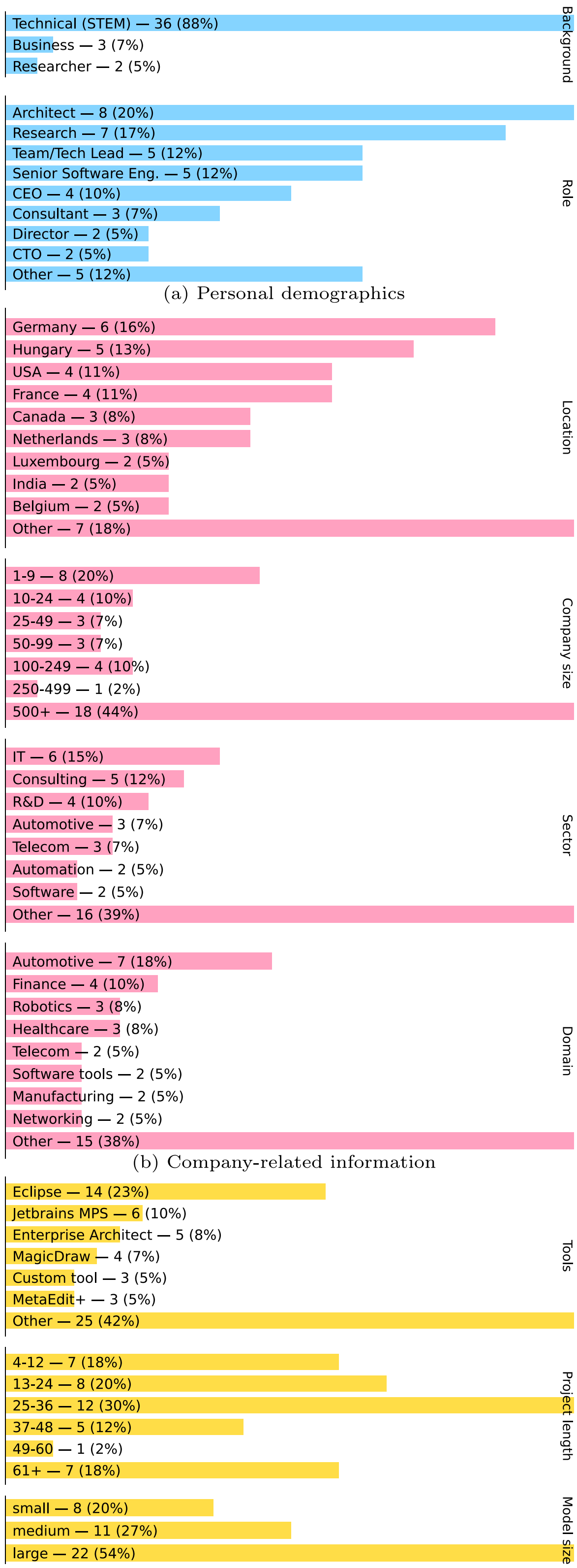
Demographics. Fig. 2 provides an overview on the main demographic information of the participants, their companies and their projects, models, and tools. About 88% of the *participants* have a primary background in STEM, 7% in business and 5% in research. The roles of the participants within their companies show a healthy mix along the corporate hierarchy, with 15% of participants being C-suite executives, 17% filling lead roles (e.g., director, team lead, head of engineering), and the rest acting in principal, senior, or architect roles. The average professional experience of participants in model-driven software engineering was around fifteen years ($\mu = 15.4, \sigma = 8.1$), with 33 of 41 participants having at least 10 years of experience. This indicates that most of our participants are experienced industry professionals working on MDSE. We aimed to reach out to participants from as many different countries as possible, however, since the researchers' location is based in Europe, the ratio of European participants in the survey is higher than others. We still have representation from other parts of the world such as North America and Asia. 44% of participants are affiliated with companies of 500+ employees, where inter-teams and intra-team collaborations can be expected on large-sized software projects. 44% of the participants are affiliated with companies below a hundred employees, but the rest of the answers to the questionnaire show that these companies are also involved in collaborative MDSE activities. The most frequently encountered *sectors* are general information technology (15%) and consulting (12%), while the most frequently encountered *application domains* were automotive (18%) and finance (10%). Eclipse and Jetbrains MPS account for 33% of the *tools or platforms* used for collaborative MDSE purposes. The projects ran from a few months (shortest being 4 months) to several years (longest being 5+ years), with about 62% of the projects ranging between 13–48 months. Over half of the participants considered their models *large*, that is, containing more than 1000 model elements. By model element, we mean the smallest unit of a model, e.g., a class in UML class diagrams. *Medium* (100–999 model elements) and *small* (less than 100 model elements) models were substantially less frequent, which was expected given that collaborative setups are more frequently encountered in large model settings. The number of collaborators in these projects ranged from as few as two people to over 500, with a mean $\mu = 39.6$ collaborators, and $\sigma = 91.4$.

Data extraction. In this phase, we collected all the answers provided by the participants of the online questionnaire into a single spreadsheet (with a column for each question and a row for each participant).

3.2.3. Phase 3: Data analysis and reporting

We analyzed the obtained Likert data both quantitatively and qualitatively. Specifically, for each question of the questionnaire (i.e., for each column of the spreadsheet containing the raw data), we applied basic descriptive statistics for a better understanding

10 https://mde-network.com
11 https://www.linkedin.com/groups/155446/



(a) Personal demographics

(b) Company-related information

(c) Project/model-related information

Fig. 2. Demographics of survey participants.

of the data about the occurrences of each given response. Following the methodological suggestions, we assume the underlying Likert data to conform to an ordinal scale (Joshi et al., 2015; Albaum, 1997; Harpe, 2015), and interpret "don't know" responses by ignoring them (Chyung et al., 2017). The number of "don't know" type responses in our data is 1.7 per question (4.1% of all responses). For space considerations, we do not report these responses in Section 4 but disclose them clearly in the replication package. In Section 4, we summarize the most frequently used collaborative MDSE techniques by ordering the single Likert items by the sum relative frequency of their positive values ("always", "often"), aggregated by the feature groups (Table 2). In Section 5, we again report relative frequencies, but this time we report overall data, not grouped by collaborative categories.

For the sake of completeness, we remark, that 162 Likert items were recorded: 81 on the practices and 81 on the needs. Although this number of items could allow combining Likert items into Likert scales, and subsequently analyzing the obtained data on more powerful scales of measurement (ratio or continuous) (Joshi et al., 2015; Harpe, 2015), our experiment was not designed to support such analysis methods, and thus, we do not consider such methods sound for the current data.

We used the open-source R framework[12] and Python for analysis. The scripts are available in the replication package.

Finally, we applied the *narrative synthesis* method for synthesizing the main findings from the extracted data. Narrative synthesis is a synthesis method whose main characteristic is the adoption of a narrative (as opposed to statistical) summary of the collected data to the process of synthesis (Rodgers et al., 2009). The narrative synthesis method is widely used in secondary studies (Cruzes and Dybå, 2011), but the descriptive nature of the design of our study makes it applicable also for our purposes. Specifically, we firstly collected the basic descriptive statistics for each question of our questionnaire and we aggregated them into a set of bar plots and tables. Then, we facilitated a series of brainstorming meetings among the authors of this paper to elaborate on and discuss the main findings. In this context, the contents of the optional open-ended questions have been used for understanding the rationale and getting additional information about their corresponding closed-ended questions. Similarly, the transcripts of the focus groups carried out in phase 1 have been used for a better understanding of the quantitative data that emerged from the online survey and for enriching the discussion of the obtained results.

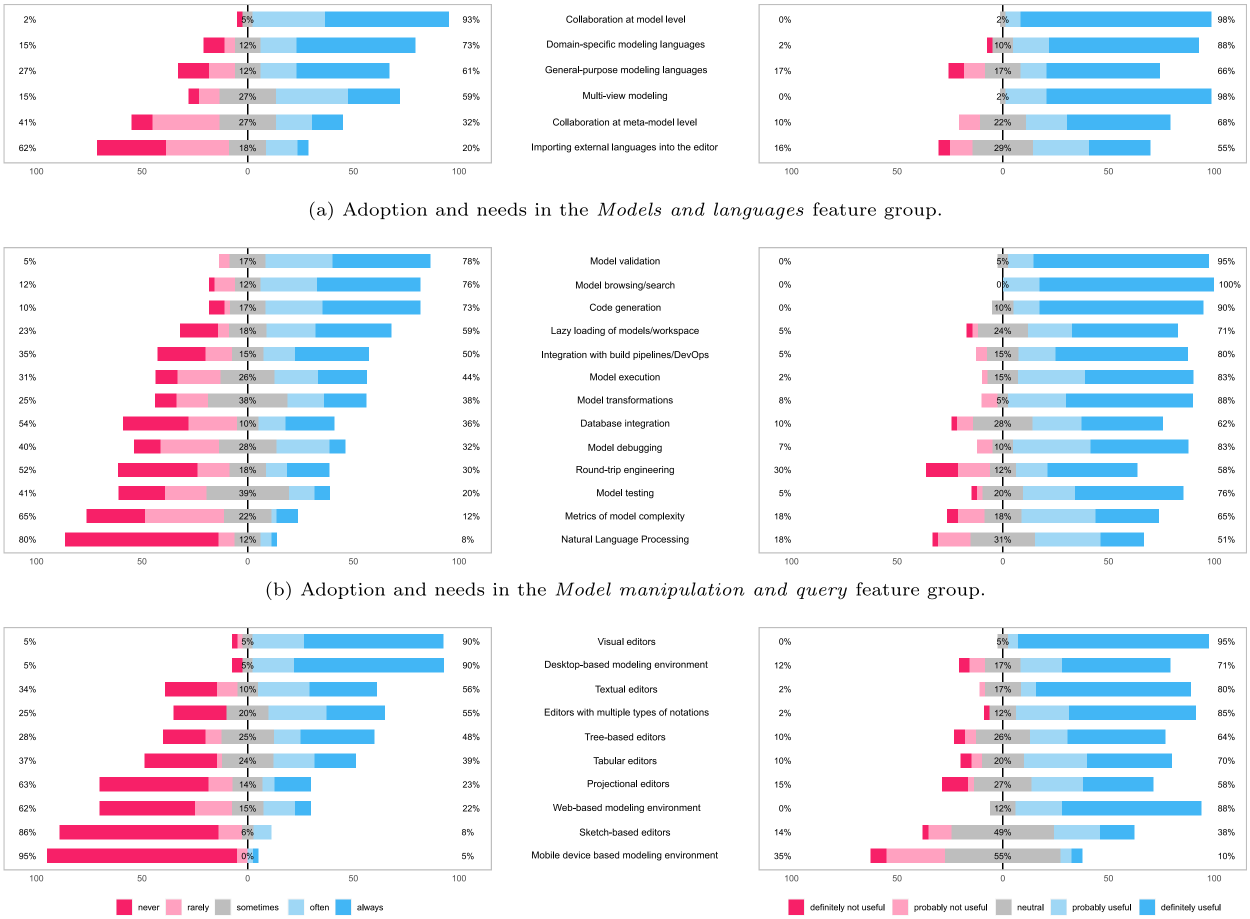
3.3. Replicability of the study

A complete replication package[1] is publicly available for independently verifying or replicating our study. The replication package includes the anonymized transcripts of the focus group sessions, the questionnaire used in the online survey, raw data of each phase of the study, the analysis scripts, together with guiding instructions about the contents of the replication package.

4. Results

In this section, we discuss the results of our study according to our classification framework for collaborative MDSE (see Table 2). For each collaborative MDSE technique, we present a pair of Likert items about (i) its current adoption by practitioners, and (ii) its expected need in future projects (see Figs. 3–5). The data is grouped by feature groups, and ordered by current adoption (i.e., the sum of the frequency of `often` and `always` values, as explained in Section 3). Detailed data is available in the replication package[1].

[12] https://www.r-project.org/



(a) Adoption and needs in the *Models and languages* feature group.

(b) Adoption and needs in the *Model manipulation and query* feature group.

(c) Adoption and needs in the *Editors and environments* feature group.

Fig. 3. Adoption and needs in the *Model management* dimension.

4.1. Model management

Fig. 3 shows the practices and needs reported by the participants related to the *Model management* dimension. Based on their current adoption and their need, web-based modeling environments are expected to be the most impactful development across every feature we have measured in this dimension. Multi-notation and multi-view modeling are also much sought-after.

Models and languages. 93% of participants have used *Collaboration at model level* in their projects, making it the most adopted technique in the group. However, the corresponding technique of *Collaborating at metamodel level* has only been used by 32% of participants. *Domain-specific modeling languages* are more adopted than *General-purpose modeling languages* (such as UML and CAD), in a ratio of 73% vs 61%. *Multi-view modeling*, although only used by 59% of practitioners previously, scores 98% in the needs, making it the most needed technique along with *Collaboration at model level* (98%).

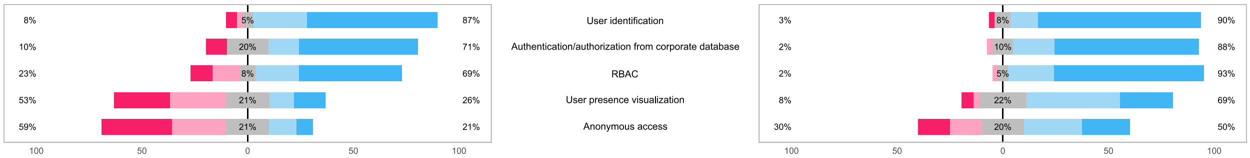
Model manipulation and query. *Model validation* (78%), *Model browsing/search* (76%) and *Code generation* (73%) are the three most adopted techniques of the group; and they score as the top three most needed features as well. Only these features score above 90% in terms of need in future projects. *Model transformations*, *Model debugging* (Van Mierlo, 2018) and *Model execution* also score high (above 80%) in the needs. Despite the recent improvements on the topic, *Natural Language Processing* (Weyssow et al., 2022; Burgueño et al., 2021) is rarely used (8%). A similarly rare technique is establishing explicit *Metrics of model complexity* (Polancic and Cegnar, 2017), which has been encountered by only 12% of participants.

Editors and modeling environments. *Visual editors* have been used by 90% of practitioners in their projects and are in high demand as well (95%). Other notations score substantially lower, with *Textual editors* at 56%, *Tree-based editors* at 48%, and *Tabular editors* at 39%. *Projectional editing* (Völter et al., 2014) has been used by only 23% of participants before. *Desktop-based modeling environments* are about four times more frequently used (90%) than *Web-based modeling environments* (22%), while *Mobile device based modeling* is sporadically adopted (5%).

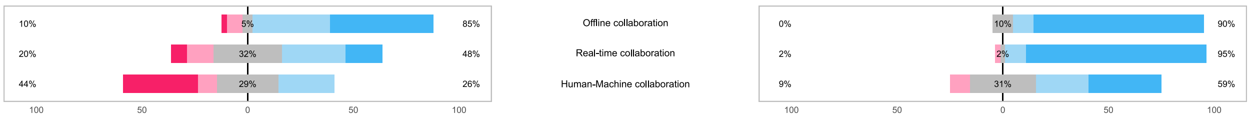
4.2. Collaboration

Fig. 4 shows the practices and needs reported by the participants related to *Collaboration*. As the main takeaway, versioning as a group scores as the most needed one across all categories (91% need on average), with every feature scoring at least 84%. Conflict awareness and the automation of conflict resolution are expected to be among the most impactful developments overall, but there is a strong need for improving the means of manual control over conflict resolution.

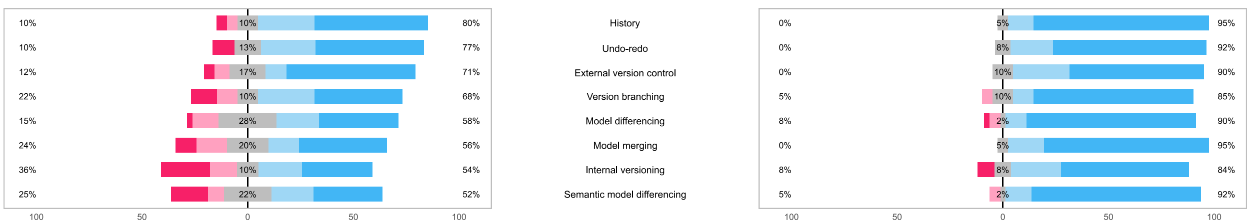
Stakeholder management and access control. 87% of the participants have used *User identification* techniques in their collaborative modeling work. *Authentication and authorization from corporate databases* (e.g., LDAP, AD) and *Role-based access control (RBAC)* are widely used as well at 71% and 69%, respectively.



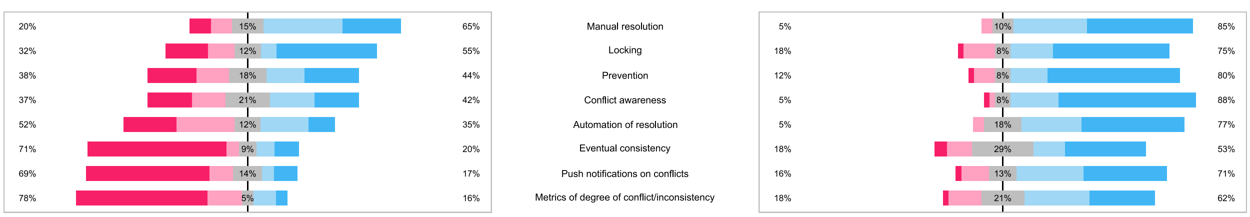
(a) Adoption and needs in the *Stakeholder management and access control* feature group.



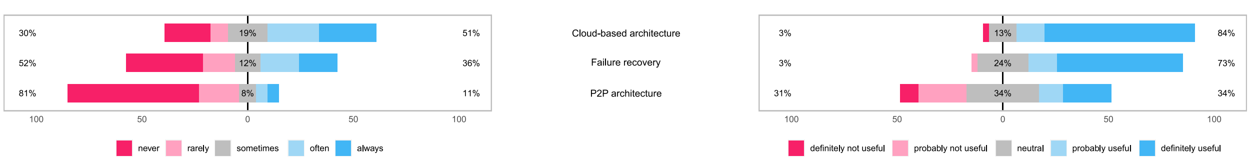
(b) Adoption and needs in the *Collaboration dynamics* feature group.



(c) Adoption and needs in the *Versioning* feature group.



(d) Adoption and needs in the *Conflicts and consistency* feature group.



(e) Adoption and needs in the *Network architecture and robustness* feature group.

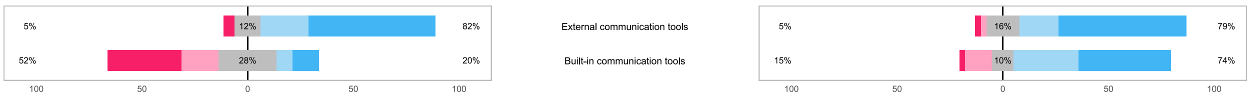
Fig. 4. Adoption and needs in the *Collaboration* dimension.

Whereas, at the other end of the scale, *Anonymous access* to shared models is a rare occurrence (21%). User presence visualization is moderately adopted at 26%. This technique, however, is much sought after.

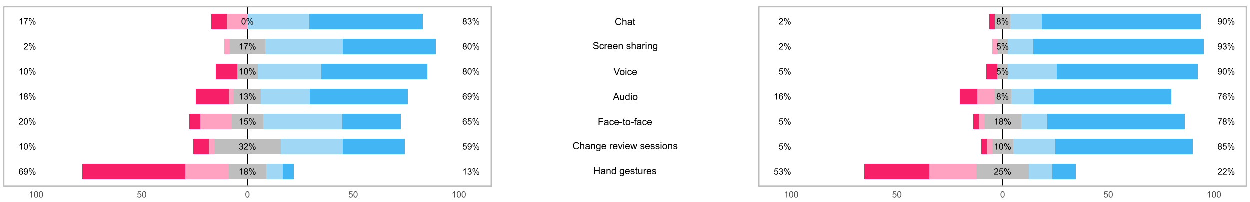
Collaboration dynamics. Most collaborative modeling endeavors run through *Offline* means of collaboration (85%), with *Real-time collaboration* being substantially less used (48%). Both collaboration types are, however, of a high need at 90% and 95%, respectively, with Real-time collaboration emerging as an impactful improvement over the current state of the practice. *Human–machine collaboration* is trending upwards, highlighted by techniques such as modeling with chatbots (Pérez-Soler et al., 2018).

Versioning. In terms of needs in future projects, this group has the highest average need at 91%, with every feature scoring at least 84%. Collaboration in software engineering has been traditionally achieved by the means of version control systems such as Git and SVN, and MDSE adopted such techniques early on (Lucia et al., 2007). 80% of participants have used some form of *History* functionality for their collaborative modeling work, and this feature is recognized as an important need (95%). *Model merging* and the required step of *Model differencing* are less adopted (56% and 58% respectively), but emerged as impactful developments of this feature group. *Internal versioning* techniques (Maróti et al., 2014b), i.e., models featuring built-in versioning mechanisms instead of relying on external tools like Git, are gaining traction, as 84% of participants agree that they are a needed addition to a collaborative modeling suite.

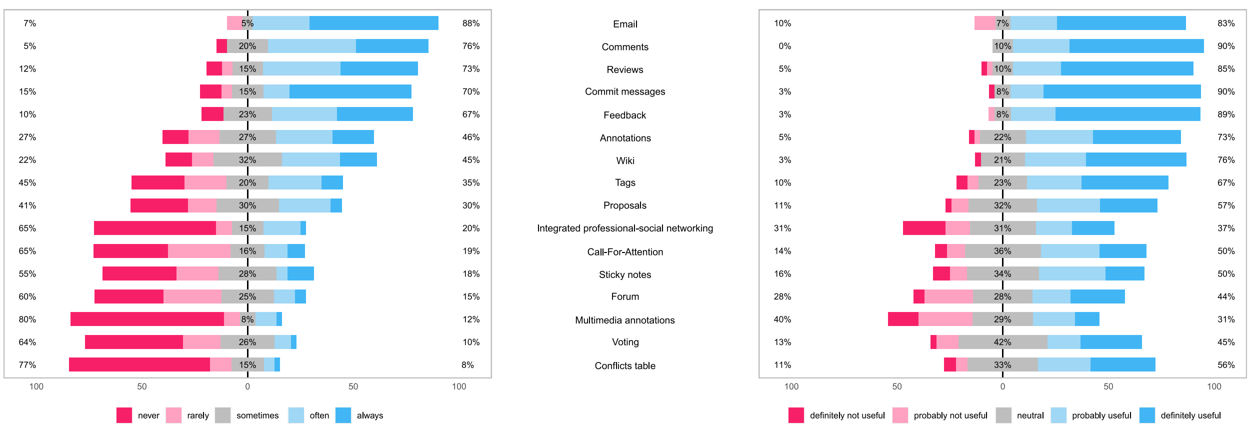
Conflicts and consistency. The recurring theme of this group is the prevalence of manual techniques. 65% of participants have worked with *Manual resolution* techniques upon encountering conflicts, making it the most adopted technique of this group. *Locking* is the second most adopted one with 55%. Only 35% of participants have used any means of *Automation of resolution*. Enabling techniques, such as *Eventual consistency* (Vogels, 2009) (20%) and *Metrics of inconsistency* (Syriani et al., 2019; David et al., 2016b) (16%) are also rarely used, similarly to *Conflict awareness* (42%) and *Notification on conflicts* (17%). Features of this group are among the least adopted ones at an average rate of 37%. However, this group projects as an impactful one, with an average need of 74%. Especially sought-after are *Conflict awareness* tools (88%



(a) Adoption and needs in the *Integration* feature group.



(b) Adoption and needs in the *Synchronous communication* feature group.



(c) Adoption and needs in the *Asynchronous communication* feature group.

Fig. 5. Adoption and needs in the *Communication* dimension.

need), improved techniques of *Manual conflict resolution* (85%), and techniques of *Conflict prevention* (80%).

Network architecture and robustness. As the main takeaways from this group, 84% of participants agree that *Cloud-based architectures* are needed in their work; and 73% find it important to implement proper *Failure recovery* techniques. Peer-to-peer architectures for collaborative modeling tend to be not adopted (11%) and they are mildly perceived as needed in future projects (34%).

4.3. Communication

Fig. 5 shows the practices and needs reported by the participants related to *Communication*. *External communication tools* are better adopted (82%) than *Built-in* ones (20%), by a wide margin. This gap, however, is substantially narrower in the need of these two approaches, projecting the development of built-in means of communication impactful.

The frequently used means of synchronous communication are *Chat* (83%), *Screen sharing* (80%) and *Voice* (80%). These techniques are also among the most needed ones. The currently employed means of asynchronous communication span across a wide range, with *Emails* being the most used (88%) and *Conflicts table* (Boulila, 2004) (8%) being the least used ones.

5. Discussion

In this section, we discuss the obtained results by (i) elaborating on the main implications related to research questions RQ1 and RQ2 (Sections Section 5.1 and (Sections Section 5.2), (ii) describing the relation between practitioners' adoption and needs of collaborative MDSE features (RQ3, Section 5.3), and (iii) presenting how the current trends of academic research align with practitioners' needs (RQ4, Section 5.4).

5.1. Adoption of collaborative MDSE features (RQ1)

Table 3 summarizes the ten most frequently and the ten least frequently used collaborative MDSE features.

Collaboration at the model level is used by nearly every participant. The only exception is related to the development of a software engineering tool in which collaboration was occasional, and mostly at the metamodel level. Models are mainly developed in *Visual editors* (90%), with the most typical supporting activities being *Model validation* (78%), and *Model browsing/search* (76%). These findings are in line with the observations of Hutchinson et al. (2014) and Akdur et al. (2018) on the most frequently used modeling languages. *Desktop-based modeling tools* are substantially more frequently adopted than *Web-based modeling tools*, at a rate of 90% to 22%. This is, however, less of a choice than a necessity: the most frequently appearing modeling tools/frameworks in our sample are Eclipse (14 occurrences; 23%), Jetbrains MPS (Pech et al., 2013) (6; 10%), Enterprise Architect[13] (5, 8%), MagicDraw (Planas and Cabot, 2020), and MetaEdit+ (Kelly and Tolvanen, 2018) (4; 7%); all of them providing a desktop-based experience. Collaboration is mostly approached in an *Offline* fashion (85%), and typically based on *External versioning* systems (71%), with clearly *Identified users* (87%). Such a collaboration model is in line with the ones observed in traditional software engineering (Mistrík et al., 2010), suggesting

13 https://sparxsystems.com/products/ea/index.html

Table 3
The ten most and least adopted and needed techniques.

(a) Adoption

Feature	Feature group	%
Collaboration at model level	Models and languages	93
Visual editors	Editors and modeling environments	90
Desktop-based environment	Editors and modeling environments	90
Email	Asynchronous communication	88
User identification	Stakeholder management	87
Offline collaboration	Collaboration dynamics	85
Chat	Synchronous communication	83
External communication tools	Integration	82
History	Versioning	80
Screen sharing	Synchronous communication	80
...		
Forum	Asynchronous communication	15
Hand gestures	Synchronous communication	13
Metrics of model complexity	Model manipulation and query	12
Multimedia annotations	Asynchronous communication	12
Peer-to-peer architecture	Network architecture and robustness	11
Voting	Asynchronous communication	10
Sketch-based editors	Editors and modeling environments	8
Conflicts table	Asynchronous communication	8
Natural Language Processing	Model manipulation and query	8
Mobile environment	Editors and modeling environments	5

(b) Need

Feature	Feature group	%
Model browsing/search	Model manipulation and query	100
Multi-view modeling	Models and languages	98
Collaboration at model level	Models and languages	98
Visual editors	Editors and modeling environments	95
Model validation	Model manipulation and query	95
History	Versioning	95
Real-time collaboration	Collaboration dynamics	95
Model merging	Versioning	95
Role-based access control	Stakeholder management	93
Screen sharing	Synchronous communication	93
...		
Call-For-Attention	Asynchronous communication	50
Anonymous access	Stakeholder management	50
Voting	Asynchronous communication	45
Forum	Asynchronous communication	44
Sketch-based editors	Editors and modeling environments	38
Integrated prof.-soc. networking	Asynchronous communication	37
Peer-to-peer architecture	Network architecture and robustness	34
Multimedia annotations	Asynchronous communication	31
Hand gestures	Synchronous communication	22
Mobile environment	Editors and modeling environments	10

that collaborative MDSE in practice is still mainly influenced by software engineering and less by model engineering. The most often used forms of Communication are *Email* and *Chat* (88% and 83%). By that, both synchronous and asynchronous means of communication appear in the top ten most frequent collaborative MDSE features; albeit not integrated with the modeling tools. Only 20% of participants used built-in means of communication. During our focus group sessions, it emerged that keeping track of the communication history about the models might be useful: *"It is not an issue per se, but it is convenient to have the model and the conversation/history of the model in the same place. Because then someone else who was not in the original Team/Zoom chat, can later read back on the discussion. Eventually, this can become important. For now, we can side-step this by Teams and email and other kinds of tools — P4"*.

Among the least adopted collaborative MDSE features are *Mobile device based modeling environments* (5%), *Natural Language Processing* (8%) and *Sketch-based editors* (8%).

5.2. Needs of collaborative MDSE features (RQ2)

Table 3 shows an excerpt of the most and the least important needs of practitioners.

Efficient and comfortable *Model browsing/search* is of an unanimous need, supported by 100% of participants, followed by *Multi-view modeling* and *Collaboration at the model level* with 98% support. *Real-time collaboration* and off-line collaboration supported by *History* both appear to be important needs, supported by 95% of participants. The already well-adopted *Visual editors* and *Model validation* features maintain their importance as well (95%). *Screen sharing* and *Role-based access control* (93%) are clearly needed collaborative supporting features. About the latter, a participant of the first focus group summarizes why role-based access control might be needed in a collaborative modeling setting: *"Role-based access control [...] is extremely important from my point of view because it has a lot to do with social issues. So people want control about the parts of the model they are responsible to. We internally are speaking about some kind of onion model alongside our department path that we always have an inner layer which is writeable by a dedicated set of users; and this inner layer is provided read-only to the next outer layer which is consuming this information. [...It] has write access to its own layer and then the next layer is always again has a kind of read access only to the second inner layer and so on and so forth. And having here very strict role-based access control avoids a lot of conflicts between the people so I think this is quite important — P2"*.

The only feature of the Communication dimension in this list, *Screen sharing*, appears with a need of 93%. As highlighted by one of the participants of FG2: *"If you have the synchronous collaboration, editing, and updating of models, then it works better if you also have a communicational connection, or even a visual connection, as we have now. [...] So then you talk about the model and you make changes, so it is a kind of collaboration where you can also use the normal, human means as talking to each other, and gestures to communicate on what you are doing. Because otherwise, it becomes very abstract: you see the same model and you see all kinds of changes without talking about it — P5"*.

Among the least needed features are *Mobile* and *Sketch-based modeling environments* (10% and 38%), and numerous types of asynchronous communication. As discussed in Section 5.3, the low need for those collaborative MDSE features can be attributed either to the novelty of such features or to the lack of usefulness of a specific feature. Future work is needed to better understand this result since our study was not designed to differentiate between these two scenarios.

5.3. Need-adoption matrix of collaborative MDSE features (RQ3)

To provide a better view of the relationship between the adoption of and the need for specific collaborative MDSE features, we chart the levels of adoption and needs against each other in what we call the Need-adoption matrix. In Fig. 6, we show three instances of the matrix — one for each collaborative dimension. Each matrix is divided into four quadrants based on how much needed and adopted certain collaborative MDSE features are. Since the Likert data of needs and adoption is not directly comparable, the matrix is meant exclusively to provide visual support about the *relative* position of the specific features with respect to their adoption and need. The Need-adoption matrix is analogous to the growth-share matrix (Henderson, 1970), colloquially known as the BCG Matrix, after its developer, the Boston Consulting Group. The *Adoption* dimension of the Need-adoption matrix aligns well with *Market share* dimension of the BCG Matrix, and the *Need* dimension aligns with *Market growth rate*.

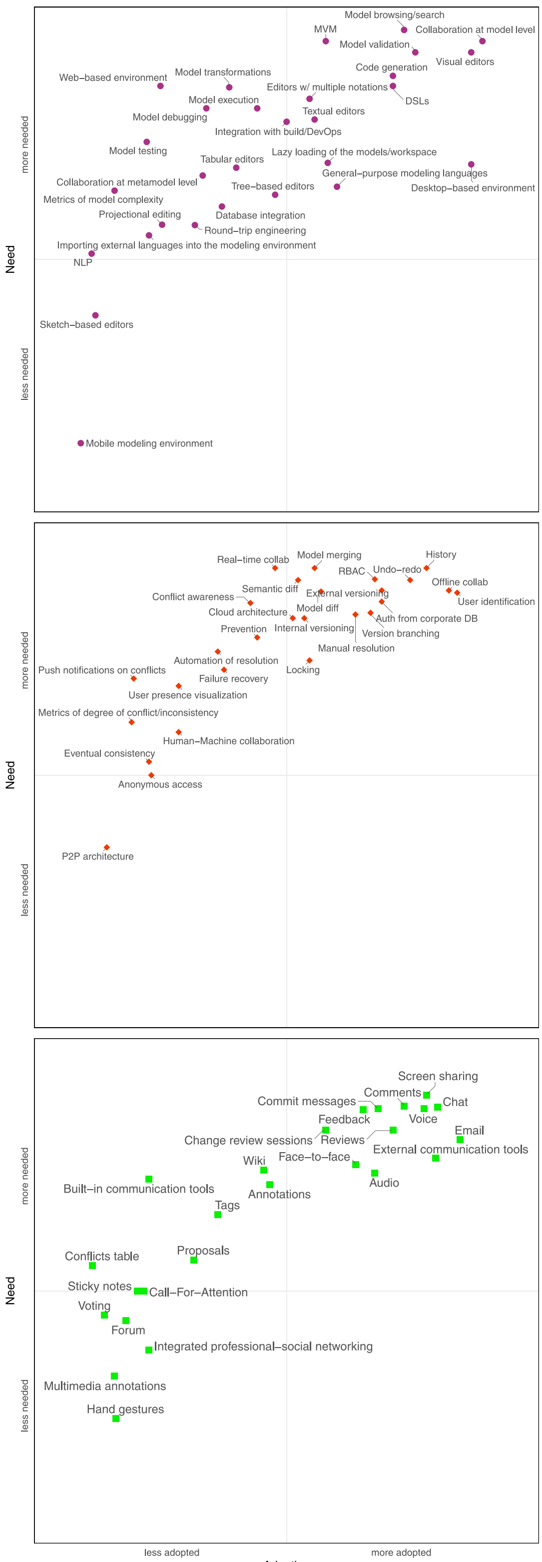


Fig. 6. Need-Adoption Matrix of the three dimensions: ● Model management, ◆ Collaboration, ■ Communication.

In the following we provide the main insights emerging from the Need-adoption matrix.[14] The following information help reason about how collaborative MDSE features are positioned relative to each other and how they should be approached in future research and development activities. While academic researchers can safely rely on the Need-adoption matrix, we recommend business stakeholders to incorporate the BCG Matrix in their decision making processes.

5.3.1. *Less needed – Less adopted*

This quadrant includes features that are either relatively new and the industry did not have the time to adopt them, or are concepts that have been known for a while, but are not needed. The features of this quadrant might have the potential to become more needed and eventually, more adopted. However, we suggest carefully analyzing each of them in their context to decide why they are situated here.

Typical activities. The main treatment of the features of this quadrant should be the exploration of potential, best situated at the border of academic research and technology transfer. Technology transfer entities and tool builders are encouraged to gauge the needs of the industry and raise awareness about emerging concepts. Researchers are encouraged to aid this process by developing prototype applications and demonstrators of their research results. The activities and their implementation align well with the *Basic research* quadrant of the Innovation Matrix (Satell, 2017).

Some characteristic examples include *Sketch-based editors* (Wüest et al., 2019) and *Mobile modeling environments* (Vaquero-Melchor et al., 2017), for which it is plausible to assume that they did not generate high interest due to their novelty; and using *Anonymous access* to modeling artifacts, which might be simply not useful in typical collaborative settings, presumably both for collaboration awareness and accountability reasons.

5.3.2. *More needed–Less adopted*

This quadrant includes collaborative MDSE features that are needed by practitioners, but they have not been significantly adopted yet. Features of this quadrant suggest excellent *break-out* potential for research, development, and industry adoption alike. Since the industry has already expressed a high interest in these features, the risks of investing effort into research and development are moderate.

Typical activities. The main treatment of the features of this quadrant should be (i) intensive research from the academic side, and (ii) rapid development from technology transfer entities and tool builders. The activities and their implementation align well with the *Breakthrough* and *Disruptive* Innovation quadrants of the Innovation Matrix (Satell, 2017).

The characteristic examples of this quadrant outline a plausible concept of the next generation of collaborative modeling tools, implemented in *Web-based environments* (Corley et al., 2016), with a combination of *Real-time* (David and Syriani, 2022; David et al., 2022b) collaboration capabilities possibly driven by *Eventual consistency models* (Vogels, 2009), and augmented by *Built-in communication tools*, with the added capabilities of *Model debugging* (Van Mierlo, 2018) and *Model testing* (Nejati et al., 2019). Also, as emerged during the first focus group session, it would be interesting to explore how *trust* among users can be ensured in a collaborative modeling setting, and here versioning and visual diff/merge might support it: *"automated conflict resolution depends on if the result is always trustable, otherwise people may get scared about it and they rather want to have manual look on it. And diff/merge is of course important, Git integration is important because people know it and want to have it somewhere in the background. They do not want to reinvent the wheel and diff visualization we will have definitely work on it together with [...], it has some nice benefits. It is not absolutely mandatory but it is worth an effort to do something there — P2".*

[14] Our sample does not record collaborative MDSE features in the *Less needed–More Adopted* quadrant, thus, we do not discuss it.

5.3.3. More needed – More adopted

This quadrant includes collaborative MDSE features that are needed by practitioners, and are already well-adopted. Features of this quadrant suggest the safest investment of effort, as the high need and wide adoption pave the way to the application of R&D results in industry.

Typical activities. The main treatment of the features of this quadrant resembles that of the *More needed–Less adopted* quadrant, but the focus is shifted from research to development. From an innovation management standpoint, these activities and their implementation align primarily with the *Sustaining Innovation* quadrant of the Innovation Matrix (Satell, 2017), suggesting incremental improvements to existing features. We remark, however, that features of this quadrant also provide opportunities for *Disruptive Innovation*, for example, by repurposing traditional software engineering techniques for collaborative MDSE. For example, a participant of our second focus group session mentioned: *"Normally if you work, for example, in a version control system, you are trying to reach a certain level of consistency before you expose [the models] to the rest of the work. So consistency on the one hand, and editing and collaboration are very important, because you do not want to expose your in-between resource which could cause errors [to other collaborating parties], it is kinda isolated. But if you do collaboration on a more detailed level, then it is fine within a small group to have an inconsistent in-between resource, and you have to agree with each other when you are going to publish it to a higher level — P5".* In this context, a potential line of research could focus on the techniques of a smooth yet sound blend of offline collaboration, real-time collaboration, (external) versioning, and model validation.

The characteristic examples of this quadrant outline a typical state-of-the-practice collaborative modeling tool, implemented in *Desktop environments* (Kelly and Tolvanen, 2018), predominantly equipped with *Visual editing* capabilities, supporting *Domain-specific modeling* (Visser, 2007); but still relying on *Offline collaboration* and the collaborative techniques inherited from traditional software engineering, such as *External version control* (SVN, Git, etc.) and *Locking* (Gómez et al., 2020); while being restricted in communication features and relying on *External communication tools*, such as *Emails* and *Commit messages*.

5.4. Alignment of academic research with needs (RQ4)

To answer RQ4, we build on the systematic mapping studies by Franzago et al. (2018) (covering research output until 2013) and David et al. (2021a) (covering research output between 2013 and 2020). We compare the data and main findings of these studies with the data collected for the current research. To this end, we map the need for specific techniques, elicited in this work, to the relative frequency of the single techniques, measured in Franzago et al. (2018) and David et al. (2021a). To allow the mapping, two researchers identified the overlap between the features of the current work (Table 2) and previous works (Franzago et al., 2018; David et al., 2021a). A third researcher validated the results. Because Franzago et al. (2018) and David et al. (2021a) guided the definition of the classification framework of this study (Table 2), the majority of the features (58 of 81 – 72%) was mapped directly. The rest of the features (23 of 81 – 28%) originate from the focus group discussions and could not be mapped directly. For the sake of replicability and independent verification, our replication package[1] contains the data from these studies, scripts for consolidating the data, and analysis scripts.

We chart the needs reported by practitioners against the frequency of research on the specific collaborative MDSE feature in Fig. 7. Similarly to Section 5.3, the charts provide visual support with *relative* values. For the correct interpretation of the matrices,

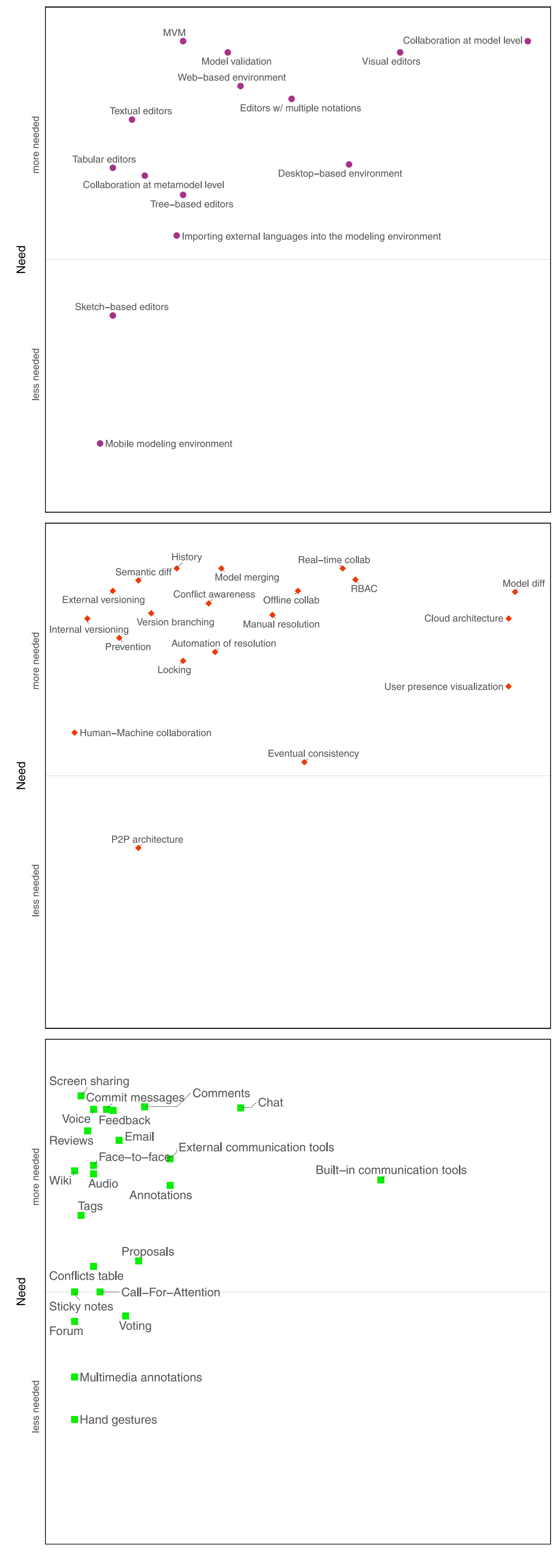


Fig. 7. Needs-Academia Matrix of the three dimensions: • Model management, ◆ Collaboration, ▪ Communication.

we note that the two axes – needs and academic intensity – are *not* measured by the same scale, and thus, their direct comparison is misleading. We aid the reader by splitting the matrices only into two halves by the need, but not by academic efforts. The following information helps academic researchers (i) to position their work with respect to industry needs, and (ii) to identify collaborative MDSE features with high application potential in industry. Prominent examples of such research areas include metamodel-level collaboration, internal version, semantic model differencing, and support for model reviews.

How to act on the findings in this section? More needed and less researched topics are obviously of a high value for researchers. Steering research towards such topics will likely receive interest from both academia and industry. The lack of industry need should not discourage prospective researchers. The lack of industry need might designate the lack of clear use-cases and motivation to adopt specific collaborative MDSE features. In such cases, transposing the results of basic research might require follow-up activities such as evangelization and technology transfer. Similarly, the lack of academic research output should not discourage industry adopters, as it might designate the lack of interest from the academic side – possibly stemming from the previously unmapped industry needs this report aims to help with – but not necessarily overly complex and unsolvable problems. In such cases, direct inquiries towards the academic community and joint research&development endeavors might be the appropriate steps (Satell, 2017).

The rest of this section is organized by the three dimensions of collaborative MDSE.

5.4.1. Model management

Every academic approach (72 of 72 – 100%) in the sample of Franzago et al. (2018) and David et al. (2021a) support *collaboration at the model level*. This result is expected since having multiple users collaborating on the same models is at the core of collaborative MDSE for both academics and survey participants. *Visual editors* (52 of 72 – 72%) are also supported by academic approaches and needed by survey participants. This might be the result of the claimed intuitiveness of visual models, especially for non-technical stakeholders (Franzago et al., 2018).

Regarding the less researched collaborative MDSE features, the most needed ones are *multi-view modeling (MVM)* (18 of 72 – 25%), *model validation* (25 of 72 – 35%), and *web-based modeling environments* (27 of 72 – 38%). However, we also observe that these three features are being studied by a fair number of approaches. For example, the management of complex MVM scenarios has been studied in Gómez et al. (2020) and Tendeloo and Vangheluwe (2018), model validation via OCL constraints in Zhu et al. (2007) and via a critics-based approach in Wieland et al. (2013), and web-based environments in WebGME (Maróti et al., 2014a) and CrowdUI (Oppenlaender et al., 2020). On the left-hand side of the top half, we observe that *textual editors* (10 of 72 – 14%) and *tabular editors* (7 of 72 – 10%) received less attention by researchers on collaborative MDSE, even though they tend to be more needed by the participants of our survey. This result is interesting since a large body of research exists, which is based on text-based language workbenches (Merkle, 2010), such as Xtext for the Eclipse Modeling Framework, but not in the context of collaborative MDSE (Erdweg et al., 2013; Eysholdt and Behrens, 2010).

Desktop-based environments (44 of 72 – 61%) are both well-supported by academic approaches and needed by survey participants. This alignment might be the result of mainstream language workbenches (e.g., Eclipse EMF, Jetbrains MPS) originating from the ecosystems of desktop-based platforms and tools. Given the growing interest in web-based editors – which is not specific to MDSE – we expect a decreasing trend in the need for desktop environments.

Features that are less needed by our survey participants and covered by only few academic approaches are *Sketch-based editors* (7 of 72 – 10%) and *Mobile modeling environments* (5 of 72 – 7%). The FlexiSketch suite (Wüest et al., 2019) and OctoUML (Vesin et al., 2017) are some of the few sketch-based modeling tools published since 2016 (David et al., 2021a). Sketch-based editors have been researched outside of the collaborative MDSE domain, e.g., by Van Mierlo et al. (2019). The situation is different in *Mobile modeling environments*, where the number of academic approaches increased to four between 2016–2020 (David et al., 2021a), while there were none before 2016 (Franzago et al., 2018). This might be the by-product of the growth of the mobile software domain, which is likely to further continue.

5.4.2. Collaboration

The Collaboration dimension shows a higher concentration of collaborative MDSE features in the top-left area (more needed, less academic approaches), suggestive of potentially valuable research topics. The leftmost side of the chart shows that model versioning techniques (*Internal versioning* and *External versioning*) (i) are the most needed feature group in the Collaboration dimension, yet (ii) are the group with less academic effort. Only three approaches (3 of 72 – 4%) for internal versioning and seven approaches (7 of 72 – 10%) for external versioning exist in the consolidated data set of Franzago et al. (2018) and David et al. (2021a). Examples of academic approaches studying external version control systems in the context of collaborative MDSE are Rocco et al. (2015), Kuryazov et al. (2018), Kelly (2017). Collaboration by internal model-based version control has been demonstrated in prototype modeling tools as well (Maróti et al., 2014b; Izquierdo and Cabot, 2016). It is important to note that advanced versioning techniques, such as domain-specific (Zadahmad et al., 2019) and semantic model differencing (Tröls et al., 2019) that leverage the semantically rich context of MDSE are trending topics in academic research (David et al., 2021a). Therefore, we expect overall growth in the versioning features on the academic side. These techniques cater to the needs of 92% of the participants of our survey who expressed *Semantic model differencing* as a need.

Real-time collaboration has been identified as the main trending topic in previous studies (David et al., 2021a; Franzago et al., 2018), aligning well with its 95% industry need. Especially driven by relaxed consistency models (such as eventual consistency (Vogels, 2009) and strong eventual consistency (Preguiça et al., 2009)), the feasibility of real-time collaborative modeling has been demonstrated in conceptual modeling (Nicolaescu et al., 2018), requirements modeling (Saini and Mussbacher, 2021), cross-platform collaborative modeling (Voogd et al., 2021), and collaborative multi-level modeling (David and Syriani, 2022). A combination of *Real-time collaboration* and traditional *Versioning* (Git) has been recently implemented in WebGME (Maróti et al., 2014b).

In terms of network architectures, both practitioners and researchers prefer *Cloud architectures* (69 of 72 – 96%) over traditional ones and over *peer-to-peer (P2P) architectures* (11 of 72 – 15%). The relatively high need and scientific interest in web-based modeling environments and the emergence of Web-based IDEs such as Eclipse Theia,[15] WebMPS,[16] etc, are suggestive of further growth in the need for cloud-based modeling environments. Recurrent reasons for adopting a peer-to-peer architecture in the academic literature are fault tolerance by avoiding having a single point of failure (Nicolaescu et al., 2018), performance improvement by direct model change messages (Sunyé, 2017), and flexibility in terms of dynamically joining collaborators (Härer, 2018).

15 https://theia-ide.org

16 https://blog.jetbrains.com/mps/2021/02/webmps-faq

5.4.3. Communication

As evidenced by Fig. 7, industry needs in the Communication dimension are in striking contrast with academic research. As concluded by the academic studies (Franzago et al., 2018; David et al., 2021a), communication features are severely overlooked by researchers of collaborative MDSE. However, our current study shows, that there is substantial industry need for such features in collaborative MDSE tools. *Screen sharing* is among the top needs (93%) and has been employed in some academic approaches as means of synchronous communication (Syriani et al., 2013; Gallardo et al., 2012; Maróti et al., 2014b). Unified model- and screen sharing has been researched as well (Tendeloo and Vangheluwe, 2018), paving the way towards novel mechanisms for much needed *Built-in communication tools* (74%).

Communication and the broader social aspects of engineering have been topics of interest in traditional Software Engineering for a long time, evidenced by long-running conference series, such as the International Conference on Cooperative and Human Aspects of Software Engineering (CHAOS)[17] and the International Conference on Global Software Engineering (ICGSE).[18] We recommend the modeling community actively looking into transposing the results of these venues. The path for such efforts has been paved by smaller, dedicated venues at the International Conference on Model Driven Engineering Languages and Systems (MODELS), including the International Workshop on Collaborative Modelling in MDE (COMMitMDE, 3 editions between 2016–2018) (Bosch et al., 2018) and the International Hands-on Workshop on Collaborative Modeling (HoWCoM, 2021) (David et al., 2021b).

In this context, *pair modeling* might be an interesting topic to investigate. We envision pair modeling following principles similar to those of pair programming in traditional software engineering — defined as two developers working side-by-side at one computer, collaborating on the same code (Williams, 2001), with the only difference that the developers are collaborating on models instead of code. It has been empirically shown that pair programming is effective in terms of quality of the produced software, duration of the programming tasks, and overall effort of developers (Hannay et al., 2009). Remote pair programming has been gaining particular popularity recently, as demonstrated by the targeted features of mainstream programming environments, such as Code With Me[19] by JetBrains and Teletype for Atom.[20] Thus, it will be interesting to investigate whether the same gains can be achieved with pair modeling in the context of collaborative MDSE.

6. Threats to validity

Construct validity. Our observations may be artifacts of the *opinions* of practitioners in our sample, rather than meaningful observations about practices and needs in the industry. To mitigate this threat, we asked participants to recall a specific *previous* MDSE project when filling in the questionnaire. The lack of exhaustiveness of the framework (Table 2) and the overlaps between its elements might be a source of additional threats to construct validity. We attempted to mitigate this effect by assembling our framework based on previous systematic studies (Franzago et al., 2018; David et al., 2021a; Masson et al., 2017). Thus, we are reasonably confident about the representativeness of the framework. Some threats might still remain as the list of supporting mechanisms for collaborative MDSE can be determined at much finer grained levels. To further mitigate threats stemming from the lack of exhaustiveness, we will maintain the framework and improve it in future work.

Internal validity. The framework we set up based on the focus group meetings and used as an input to the questionnaire (Table 2) might result in incorrect categories of concerns. Typically orthogonal concerns might be implicit in our framework, e.g., interoperability of heterogeneous modeling environments as described by Demuth et al. (2015) might appear distributed across the concerns of editors supporting multiple types of notations, integration with build and DevOps tools and databases, network architecture, etc. Furthermore, because of the sometimes broad definition of the categories, there is room for interpretation when answering the questionnaire. Overloaded terms and ambiguous concepts – e.g., model validation – might be sources of threats to internal validity as well. To address these threats, we validated the framework against previous mapping studies (Franzago et al., 2018; David et al., 2021a) and feature models (Masson et al., 2017), and provided the participants with definitions of the main dimensions of the framework.

External validity. The number of participants in our study is a substantial threat to external validity and the safe generalization of results is likely not possible. Our study sampled 41 participants who have industry experience in collaborative MDSE. To improve the representativeness of our data, we attempted to maximize the number of *companies* in our sample by restricting the number of participants to one per company whenever possible. This was possible in the case of directly recruited participants. However, due to the anonymity of the survey, we cannot exclude the possibility of multiple people being present in our sample from the same company. We estimate that the 41 participants are employed by 38 different companies. Still, these figures do not allow for safe generalization. However, the goal of this study was not to provide a general theory for collaborative MDSE but to extract high-level insights from authentic sources that are hard to access for academists, and to identify the main gaps in the practices and needs in collaborative MDSE in industry.

Conclusion validity. The Likert data in our results does not constitute a Likert scale and therefore, it cannot be soundly analyzed in a purely quantitative fashion (Albaum, 1997). Standalone and independent Likert items, such as the ones in our study, can be safely analyzed by modes, medians, and frequencies (Harpe, 2015). Due to the large number of elements in our framework (Table 2) not allowing for a concise discussion of the results in these term, we decided to base the discussion on percentage-scaled Likert data. However, this choice might lead to threats to conclusion validity. We applied two countermeasures to mitigate these threats and improve the validity of the conclusions drawn in this paper. First, we considered the percentage values as ordinal indicators of adoption and need rather than ratio-scale values. That is, the difference between 40% and 50% of need might not be the same as the difference between 90% and 100%. This is especially important when considering the diminishing difference between need and current adoption as adoption converges to 100%. Second, in a related step, we constructed the Need-adoption matrix which allows for the visual inspection and interpretation of stratified relations between adoption and need.

7. Related work

In this section, we discuss the related work: surveys and secondary studies with an adjacent scope to our current work. Specifically, we look into surveys on collaborative SE (collaborative, but not model-driven and not practitioner-oriented);

17 https://conf.researchr.org/home/chase-2021
18 https://conf.researchr.org/series/icgse
19 https://www.jetbrains.com/code-with-me/
20 https://teletype.atom.io/

practitioner surveys on general MDSE (model-driven, practitioner-oriented, but not collaborative); and surveys on collaborative MDSE (collaborative, model-driven, but not practitioner-oriented).

Surveys on collaborative SE. The increasing trend of software development using globally distributed teams introduces collaboration difficulties. The problems of dispersed team members having to interact across the hurdles of different time zones, languages, and cultures have been studied extensively (Carmel and Agarwal, 2001; Taxén, 2006; Espinosa and Carmel, 2003). The investigation about whom software engineers collaborate with and how that collaboration is performed on tasks at a large software company is conducted through interviews in Ford et al. (2017). Further exploration of the factors affecting effective collaboration was studied through questionnaire-guided interviews in globally distributed software development projects (Mohapatra et al., 2010). Collaboration during specific phases of software development lifecycle, such as software design, is studied through a set of interviews conducted with software architects working at a large global software solutions provider (Bang et al., 2010). In our work, we combined focus group study and questionnaire-based survey to particularly explore the collaboration among practitioners where models are the collaboration subject.

Practitioner surveys on general MDSE. There is a vast body of knowledge on the practices and needs of industry in *general* MDSE, that is, without an emphasis on collaboration. Close to our methods, a questionnaire-based survey was performed in combination with focus group studies by Mirri et al. (2018) to investigate if a higher level of satisfaction can be achieved by involving end-users in the software design process, as compared to the traditional design process. At the end of the software prototypes development, an evaluation was performed with target users and it was found that the prototype developed with user involvement received a higher satisfaction score. Interviews, conducted both in-person and online, with software architects of a large global software company were found effective to understand collaborative software design practices by Bang et al. (2010). The study identified roles, collaboration patterns, topologies, and geographical distribution of software architects as well as the factors impacting cost in collaborative software design. Awotunde et al. (2020) use questionnaires and message logs to investigate the impact of communication among stakeholders on the software development process. The study was performed by analyzing the communication during an android app and website development project between final year bachelor students (developers) and their lecturers (project managers). The study is performed in an academic setting and has the threat of not being able to reveal the challenges faced in communication in an industrial setting. Surveys and interviews were also used in the past to evaluate the practices and needs of model-driven software engineering in industry (Akdur et al., 2018; Mohagheghi et al., 2013; Hutchinson et al., 2011).

Surveys on collaborative MDSE. The state of the art in collaborative MDSE has been systematically assessed in previous studies (Franzago et al., 2018; David et al., 2021a). These studies, however, focus on mapping the main characteristics, challenges, and publications trends of topical academic research, and leave room for improvement, including the mapping and classification of needs and practices in industry. We constructed our classification framework based on these studies, and augmented it with 23 more collaborative MDSE features from the focus group phase of our study. Further related studies on collaborative MDSE have been carried out by Masson et al. (2017) on the features of collaborative modeling tools; and Stephan (Stephan, 2019) on the emerging keywords in collaborative MDSE between 2012–2017.

8. Conclusions and future work

In this paper, we have reported the results of our survey on the practices and needs of industry in collaborative model-driven software engineering. Based on focus group discussions with industrial experts, and an online questionnaire survey, we have obtained valuable data that we analyzed both quantitatively and qualitatively. Our study reveals frequently encountered and sought-after features of collaborative MDSE in industry settings. In addition, we have assessed how academic research aligns with these needs. Our study provides current and prospective academic researchers with firm leads to appropriately steer their research. Topics with elevated research upside include collaborative multi-view modeling and web-based collaboration. Industry practitioners and tool builders can benefit from the findings of this paper by anticipating the next generation of collaborative MDSE tools and preparing for the associated challenges. Such features include collaboration at meta-model levels and better support for communication facilities — both of which give rise to unique challenges. This paper reports only the most essential findings. More insights can be gained from our publicly available and partially pre-processed data set.[1] In future work, we plan to maintain and gradually improve the framework presented in this paper. This will enable us to carry out a tool survey and catalog currently available collaborative MDSE tools along with their feature model. A similar approach has been followed for classification the related field of model version control by Altmanninger et al. (2009).

CRediT authorship contribution statement

Istvan David: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing. **Kousar Aslam:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing. **Ivano Malavolta:** Conceptualization, Methodology, Validation, Investigation, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition. **Patricia Lago:** Conceptualization, Methodology, Validation, Investigation, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Associated data is available on Zenodo. URL and DOI available in the paper.

Acknowledgments

The authors would like to thank the participants of the focus group sessions and the online questionnaire survey for their expert input and constructive feedback. This research was partially funded by the Rijksdienst voor Ondernemend Nederland (RVO) through the ITEA3 BUMBLE project (18006).

References

Akdur, D., Garousi, V., Demirörs, O., 2018. A survey on modeling and model-driven engineering practices in the embedded software industry. J. Syst. Archit. 91, 62–82. http://dx.doi.org/10.1016/j.sysarc.2018.09.007.

Albaum, G., 1997. The likert scale revisited. Mark. Res. Society. Journal. 39 (2), 1–21. http://dx.doi.org/10.1177/147078539703900202.

Altmanninger, K., Seidl, M., Wimmer, M., 2009. A survey on model versioning approaches. Int. J. Web Inf. Syst. 5 (3), 271–304. http://dx.doi.org/10.1108/17440080910983556.

Awotunde, J.B., Ayo, F.E., Ogundokun, R.O., Matiluko, O.E., Adeniyi, E.A., 2020. Investigating the roles of effective communication among stakeholders in collaborative software development projects. In: Computational Science and Its Applications – 20th International Conference, Cagliari, Italy, Proceedings, Part VI. In: LNCS, vol. 12254, Springer, pp. 311–319. http://dx.doi.org/10.1007/978-3-030-58817-5_24.

Bang, J.Y., Brun, Y., Medvidovic, N., 2017. Continuous analysis of collaborative design. In: 2017 IEEE International Conference on Software Architecture, Gothenburg, Sweden, 2017. IEEE, pp. 97–106. http://dx.doi.org/10.1109/ICSA.2017.45.

Bang, J.Y., Popescu, D., Edwards, G., Medvidovic, N., Kulkarni, N.N., Rama, G.M., Padmanabhuni, S., 2010. Codesign: a highly extensible collaborative software modeling framework. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, Cape Town, South Africa. ACM, pp. 243–246. http://dx.doi.org/10.1145/1810295.1810341.

Bosch, J., Gerard, S., Kolovos, D.S., Malavolta, I., Muccini, H., 2018. COMMitMDE 2018 - 3rd International workshop on collaborative modelling in MDE. In: Proceedings of MODELS 2018 Workshops Co-Located with ACM/IEEE 21st International Conference on Model Driven Engineering Languages and Systems, Copenhagen, Denmark. In: CEUR Workshop Proceedings, vol. 2245, CEUR-WS.org, pp. 266–267.

Boulila, N., 2004. Group support for distributed collaborative concurrent software modeling. In: 19th IEEE International Conference on Automated Software Engineering, Linz, Austria. IEEE, pp. 422–425. http://dx.doi.org/10.1109/ASE.2004.10014.

Brambilla, M., Cabot, J., Wimmer, M., 2017. Model-Driven Software Engineering in Practice, Second Edition. In: Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers, http://dx.doi.org/10.2200/S00751ED2V01Y201701SWE004.

Brosch, P., Seidl, M., Wimmer, M., Kappel, G., 2012. Conflict visualization for evolving UML models. J. Object Technol. 11 (3), 2: 1–30. http://dx.doi.org/10.5381/jot.2012.11.3.a2.

Bucchiarone, A., Ciccozzi, F., Lambers, L., Pierantonio, A., Tichy, M., Tisi, M., Wortmann, A., Zaytsev, V., 2021. What is the future of modeling? IEEE Softw. 38 (2), 119–127. http://dx.doi.org/10.1109/MS.2020.3041522.

Burgueño, L., Clarisó, R., Gérard, S., Li, S., Cabot, J., 2021. An NLP-based architecture for the autocompletion of partial domain models. In: Advanced Information Systems Engineering - 33rd International Conference, Melbourne, VIC, Australia. In: LNCS, vol. 12751, Springer, pp. 91–106. http://dx.doi.org/10.1007/978-3-030-79382-1_6.

Cabot, J., 2020. Positioning of the low-code movement within the field of model-driven engineering. In: MODELS '20: ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems, Virtual Event, Canada, Companion Proceedings. ACM, pp. 76:1–76:3. http://dx.doi.org/10.1145/3417990.3420210.

Carmel, E., Agarwal, R., 2001. Tactical approaches for alleviating distance in global software development. IEEE Softw. 18 (2), 22–29. http://dx.doi.org/10.1109/52.914734.

Chyung, S.Y.Y., Roberts, K., Swanson, I., Hankinson, A., 2017. Evidence-based survey design: The use of a midpoint on the likert scale. Perform. Improvement 56 (10), 15–23. http://dx.doi.org/10.1002/pfi.21727.

Corley, J., Syriani, E., Ergin, H., Van Mierlo, S., 2016. Cloud-based multi-view modeling environments. In: Modern Software Engineering Methodologies for Mobile and Cloud Environments. IGI Global, pp. 120–139. http://dx.doi.org/10.4018/978-1-4666-9916-8.ch007.

Cruzes, D.S., Dybå, T., 2011. Research synthesis in software engineering: A tertiary study. Inf. Softw. Technol. 53 (5), 440–455. http://dx.doi.org/10.1016/j.infsof.2011.01.004.

David, I., Aslam, K., Faridmoayer, S., Malavolta, I., Syriani, E., Lago, P., 2021a. Collaborative model-driven software engineering: A systematic update. In: 24th International Conference on Model Driven Engineering Languages and Systems, MODELS 2021, Fukuoka, Japan, October 10-15, 2021. IEEE, pp. 273–284. http://dx.doi.org/10.1109/MODELS50736.2021.00035.

David, I., Denil, J., Gadeyne, K., Vangheluwe, H., 2016a. Engineering process transformation to manage (in)consistency. In: Proceedings of the 1st International Workshop on Collaborative Modelling in MDE Co-Located with ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, St. Malo, France. In: CEUR Workshop Proceedings, vol. 1717, CEUR-WS.org, pp. 7–16.

David, I., Latifaj, M., Pietron, J., Zhang, W., Ciccozzi, F., Malavolta, I., Raschke, A., Steghöfer, J.-P., Hebig, R., 2022a. Blended modeling in commercial and open-source model-driven software engineering tools: A systematic study. Softw. Syst. Model. http://dx.doi.org/10.1007/s10270-022-01010-3.

David, I., Syriani, E., 2022. Real-time collaborative multi-level modeling by conflict-free replicated data types. Softw. Syst. Model. http://dx.doi.org/10.1007/s10270-022-01054-5.

David, I., Syriani, E., García-Domínguez, A., 2021b. Preface to the 1st international hands-on workshop on collaborative modeling (HoWCoM 2021). In: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, Fukuoka, Japan. IEEE, pp. 1–2. http://dx.doi.org/10.1109/MODELS-C53483.2021.00001.

David, I., Syriani, E., Masson, C., 2022b. Extensible conflict-free replicated datatypes for real-time collaborative software engineering. In: Proceedings of the 17th Conference on Computer Science and Intelligence Systems, Sofia, Bulgaria. In: Annals of Computer Science and Information Systems, vol. 30, pp. 849–853. http://dx.doi.org/10.15439/2022F99.

David, I., Syriani, E., Verbrugge, C., Buchs, D., Blouin, D., Cicchetti, A., Vanherpen, K., 2016b. Towards inconsistency tolerance by quantification of semantic inconsistencies. In: Proceedings of the 1st International Workshop on Collaborative Modelling in MDE Co-Located with ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, St. Malo, France. In: CEUR Workshop Proceedings, vol. 1717, CEUR-WS.org, pp. 35–44.

Demuth, A., Riedl-Ehrenleitner, M., Nöhrer, A., Hehenberger, P., Zeman, K., Egyed, A., 2015. DesignSpace: an infrastructure for multi-user/multi-tool engineering. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain. ACM, pp. 1486–1491. http://dx.doi.org/10.1145/2695664.2695697.

Erdweg, S., et al., 2013. The state of the art in language workbenches - conclusions from the language workbench challenge. In: Software Language Engineering - 6th International Conference, SLE 2013, Indianapolis, in, USA. Proceedings. In: LNCS, vol. 8225, Springer, pp. 197–217. http://dx.doi.org/10.1007/978-3-319-02654-1_11.

Espinosa, J.A., Carmel, E., 2003. The impact of time separation on coordination in global software teams: a conceptual foundation. Softw. Process. Improv. Pract. 8 (4), 249–266. http://dx.doi.org/10.1002/spip.185.

Eysholdt, M., Behrens, H., 2010. Xtext: implement your language faster than the quick and dirty way. In: Companion to the 25th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2010, Part of SPLASH 2010, Reno/Tahoe, Nevada, USA. ACM, pp. 307–309. http://dx.doi.org/10.1145/1869542.1869625.

Ford, D., Zimmermann, T., Bird, C., Nagappan, N., 2017. Characterizing software engineering work with personas based on knowledge worker actions. In: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Toronto, on, Canada. IEEE, pp. 394–403. http://dx.doi.org/10.1109/ESEM.2017.54.

Franzago, M., Ruscio, D.D., Malavolta, I., Muccini, H., 2018. Collaborative model-driven software engineering: A classification framework and a research map. IEEE Trans. Software Eng. 44 (12), 1146–1175. http://dx.doi.org/10.1109/TSE.2017.2755039.

Gallardo, J., Bravo, C., Redondo, M.A., 2012. A model-driven development method for collaborative modeling tools. J. Netw. Comput. Appl. 35 (3), 1086–1105. http://dx.doi.org/10.1016/j.jnca.2011.12.009.

Gómez, A., Mendialdua, X., Barmpis, K., Bergmann, G., Cabot, J., Carlos, X.D., Debreceni, C., Garmendia, A., Kolovos, D.S., de Lara, J., 2020. Scalable modeling technologies in the wild: an experience report on wind turbines control applications development. Softw. Syst. Model. 19 (5), 1229–1261. http://dx.doi.org/10.1007/s10270-020-00776-8.

Hannay, J.E., Dybå, T., Arisholm, E., Sjøberg, D.I.K., 2009. The effectiveness of pair programming: A meta-analysis. Inf. Softw. Technol. 51 (7), 1110–1122. http://dx.doi.org/10.1016/j.infsof.2009.02.001.

Härer, F., 2018. Decentralized business process modeling and instance tracking secured by a blockchain. In: 26th European Conference on Information Systems: Beyond Digitization – Facets of Socio-Technical Change, Portsmouth, UK. p. 55.

Harpe, S.E., 2015. How to analyze likert and other rating scale data. Curr. Pharmacy Teach. Learn. 7 (6), 836–850. http://dx.doi.org/10.1016/j.cptl.2015.08.001.

Henderson, B.D., 1970. The product portfolio. Boston Consult. Group Perspect. 66.

Herbsleb, J.D., 2007. Global software engineering: The future of socio-technical coordination. In: International Conference on Software Engineering, ISCE 2007, Workshop on the Future of Software Engineering, Minneapolis, MN, USA. IEEE, pp. 188–198. http://dx.doi.org/10.1109/FOSE.2007.11.

Hutchinson, J.E., Rouncefield, M., Whittle, J., 2011. Model-driven engineering practices in industry. In: Proceedings of the 33rd International Conference on Software Engineering, Waikiki, Honolulu , HI, USA. ACM, pp. 633–642. http://dx.doi.org/10.1145/1985793.1985882.

Hutchinson, J.E., Whittle, J., Rouncefield, M., 2014. Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. Sci. Comput. Program. 89, 144–161. http://dx.doi.org/10.1016/j.scico.2013.03.017.

Izquierdo, J.L.C., Cabot, J., 2016. Collaboro: a collaborative (meta) modeling tool. PeerJ Comput. Sci. 2, e84. http://dx.doi.org/10.7717/peerj-cs.84.

Joshi, A., et al., 2015. Likert scale: Explored and explained. Br. J. Appl. Sci. Technol. 7 (4), 396.

Kehrer, T., Kelter, U., Taentzer, G., 2011. A rule-based approach to the semantic lifting of model differences in the context of model versioning. In: 26th IEEE/ACM International Conference on Automated Software Engineering, Lawrence, KS, USA. IEEE, pp. 163–172. http://dx.doi.org/10.1109/ASE.2011.6100050.

Kelly, S., 2017. Collaborative modelling with version control. In: Software Technologies: Applications and Foundations - STAF 2017 Collocated Workshops, Marburg, Germany, July 17-21, 2017, Revised Selected Papers. In: LNCS, vol. 10748, Springer, pp. 20–29. http://dx.doi.org/10.1007/978-3-319-74730-9_3.

Kelly, S., Tolvanen, J., 2018. Collaborative creation and versioning of modeling languages with MetaEdit+. In: Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Copenhagen, Denmark. ACM, pp. 37–41. http://dx.doi.org/10.1145/3270112.3270132.

Kitchenham, B.A., Pfleeger, S.L., 2002. Principles of survey research: part 5: populations and samples. ACM SIGSOFT Softw. Eng. Notes 27 (5), 17–20. http://dx.doi.org/10.1145/571681.571686.

Kolovos, D.S., Rose, L.M., Matragkas, N.D., Paige, R.F., Guerra, E., Cuadrado, J.S., de Lara, J., Ráth, I., Varró, D., Tisi, M., Cabot, J., 2013. A research roadmap towards achieving scalability in model driven engineering. In: Proceedings of the Workshop on Scalability in Model Driven Engineering, Budapest, Hungary. ACM, p. 2. http://dx.doi.org/10.1145/2487766.2487768.

Kontio, J., Bragge, J., Lehtola, L., 2008. The focus group method as an empirical tool in software engineering. In: Guide to Advanced Empirical Software Engineering. Springer London, pp. 93–116. http://dx.doi.org/10.1007/978-1-84800-044-5_4.

Kontio, J., Lehtola, L., Bragge, J., 2004. Using the focus group method in software engineering: Obtaining practitioner and user experiences. In: 2004 International Symposium on Empirical Software Engineering (ISESE 2004), 2004, Redondo Beach, CA, USA. IEEE, pp. 271–280. http://dx.doi.org/10.1109/ISESE.2004.35.

Kuryazov, D., Winter, A., Reussner, R.H., 2018. Collaborative modeling enabled by version control. In: Modellierung 2018, Braunschweig, Germany. In: LNI, In: LNI, vol. P-280.183–198,

Latifaj, M., Ciccozzi, F., Anwar, M.W., Mohlin, M., 2021. Blended graphical and textual modelling of UML-RT state-machines: An industrial experience. In: Software Architecture - 15th European Conference, Tracks and Workshops; VäXjÖ, Sweden, Revised Selected Papers. In: LNCS, 13365, Springer, pp. 22–44. http://dx.doi.org/10.1007/978-3-031-15116-3_2.

Lucia, A.D., Fasano, F., Scanniello, G., Tortora, G., 2007. Enhancing collaborative synchronous UML modelling with fine-grained versioning of software artefacts. J. Vis. Lang. Comput. 18 (5), 492–503. http://dx.doi.org/10.1016/j.jvlc.2007.08.005.

Maróti, M., Kecskés, T., Kereskényi, R., Broll, B., Völgyesi, P., Jurácz, L., Levendovszky, T., Lédeczi, Á., 2014a. Next generation (meta)modeling: Web- and cloud-based collaborative tool infrastructure. In: Proceedings of the 8th Workshop on Multi-Paradigm Modeling Co-Located with the 17th International Conference on Model Driven Engineering Languages and Systems, Valencia, Spain. In: CEUR Workshop Proceedings, In: CEUR Workshop Proceedings, 1237.41–60,

Maróti, M., Kereskényi, R., Kecskés, T., Völgyesi, P., Lédeczi, Á., 2014b. Online collaborative environment for designing complex computational systems. In: Proceedings of the International Conference on Computational Science, Cairns, Queensland, Australia. In: Procedia Computer Science, In: Procedia Computer Science, vol. 29, 2432–2441.http://dx.doi.org/10.1016/j.procs.2014.05.227,

Masson, C., Corley, J., Syriani, E., 2017. Feature model for collaborative modeling environments. In: Proceedings of MODELS 2017 Satellite Events, Austin, TX, USA, September, 17, 2017. In: CEUR Workshop Proceedings, vol. 2019, CEUR-WS.org, pp. 164–173.

Merkle, B., 2010. Textual modeling tools: overview and comparison of language workbenches. In: Cook, W.R., Clarke, S., Rinard, M.C. (Eds.), Companion To the 25th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2010, Part of SPLASH 2010, Reno/Tahoe, Nevada, USA. ACM, pp. 139–148. http://dx.doi.org/10.1145/1869542.1869564.

Merriam, S.B., Tisdell, E.J., 2015. Qualitative Research: A Guide To Design and Implementation. John Wiley & Sons.

Mirri, S., Roccetti, M., Salomoni, P., 2018. Collaborative design of software applications: the role of users. Hum. Centric Comput. Inf. Sci. 8, 6. http://dx.doi.org/10.1186/s13673-018-0129-6.

Mistrík, I., Grundy, J., van der Hoek, A., Whitehead, J., 2010. Collaborative software engineering: Challenges and prospects. In: Collaborative Software Engineering. Springer, pp. 389–403. http://dx.doi.org/10.1007/978-3-642-10294-3_19.

Mohagheghi, P., Gilani, W., Stefanescu, A., Fernández, M.A., 2013. An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases. Empir. Softw. Eng. 18 (1), 89–116. http://dx.doi.org/10.1007/s10664-012-9196-x.

Mohapatra, P., Björndal, P., Smiley, K., 2010. Causal analysis of factors governing collaboration in global software development teams. In: 5th IEEE International Conference on Global Software Engineering, Princeton, NJ, USA. IEEE, pp. 128–132. http://dx.doi.org/10.1109/ICGSE.2010.23.

Muccini, H., Bosch, J., van der Hoek, A., 2018. Collaborative modeling in software engineering. IEEE Softw. 35 (6), 20–24. http://dx.doi.org/10.1109/MS.2018.4321244.

Mussbacher, G., et al., 2020. Opportunities in intelligent modeling assistance. Softw. Syst. Model. 19 (5), 1045–1053. http://dx.doi.org/10.1007/s10270-020-00814-5.

Nejati, S., Gaaloul, K., Menghi, C., Briand, L.C., Foster, S., Wolfe, D., 2019. Evaluating model testing and model checking for finding requirements violations in simulink models. In: Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia. ACM, pp. 1015–1025. http://dx.doi.org/10.1145/3338906.3340444.

Nicolaescu, P., Rosenstengel, M., Derntl, M., Klamma, R., Jarke, M., 2018. Near real-time collaborative modeling for view-based web information systems engineering. Inf. Syst. 74 (Part), 23–39. http://dx.doi.org/10.1016/j.is.2017.07.008.

Oliver, D.G., Serovich, J.M., Mason, T.L., 2005. Constraints and opportunities with interview transcription: Towards reflection in qualitative research. Soc. Forces 84 (2), 1273–1289. http://dx.doi.org/10.1353/sof.2006.0023.

Oppenheim, A.N., 2000. Questionnaire Design, Interviewing and Attitude Measurement. Bloomsbury Publishing.

Oppenlaender, J., Tiropanis, T., Hosio, S., 2020. CrowdUI: Supporting web design with the crowd. Proc. ACM Hum. Comput. Interact. 4 (EICS), 76:1–76:28. http://dx.doi.org/10.1145/3394978.

Pech, V., Shatalin, A., Voelter, M., 2013. JetBrains MPS as a tool for extending java. In: Proceedings of the 2013 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools, Stuttgart, Germany. ACM, pp. 165–168. http://dx.doi.org/10.1145/2500828.2500846.

Pérez-Soler, S., Guerra, E., de Lara, J., 2018. Collaborative modeling and group decision making using chatbots in social networks. IEEE Softw. 35 (6), 48–54. http://dx.doi.org/10.1109/MS.2018.290101511.

Persson, M., Törngren, M., Qamar, A., Westman, J., Biehl, M., Tripakis, S., Vangheluwe, H., Denil, J., 2013. A characterization of integrated multi-view modeling in the context of embedded and cyber-physical systems. In: Proceedings of the International Conference on Embedded Software, Montreal, QC, Canada. IEEE, pp. 10:1–10:10. http://dx.doi.org/10.1109/EMSOFT.2013.6658588.

Planas, E., Cabot, J., 2020. How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and papyrus. Comput. Stand. Interfaces 67, http://dx.doi.org/10.1016/j.csi.2019.103363.

Polancic, G., Cegnar, B., 2017. Complexity metrics for process models - A systematic literature review. Comput. Stand. Interfaces 51, 104–117. http://dx.doi.org/10.1016/j.csi.2016.12.003.

Preguiça, N.M., Marquès, J.M., Shapiro, M., Letia, M., 2009. A commutative replicated data type for cooperative editing. In: 29th IEEE International Conference on Distributed Computing Systems, Montreal, QC, Canada. IEEE, pp. 395–403. http://dx.doi.org/10.1109/ICDCS.2009.20.

Rocco, J.D., Ruscio, D.D., Iovino, L., Pierantonio, A., 2015. Collaborative repositories in model-driven engineering. IEEE Softw. 32 (3), 28–34. http://dx.doi.org/10.1109/MS.2015.61.

Rodgers, M., Sowden, A., Petticrew, M., Arai, L., Roberts, H., Britten, N., Popay, J., 2009. Testing methodological guidance on the conduct of narrative synthesis in systematic reviews. Evaluation 15 (1), 49–73. http://dx.doi.org/10.1177/1356389008097871.

Saini, R., Mussbacher, G., 2021. Towards conflict-free collaborative modelling using VS code extensions. In: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, Fukuoka, Japan. IEEE, pp. 35–44. http://dx.doi.org/10.1109/MODELS-C53483.2021.00013.

Sanchis, R., García-Perales, Ó., Fraile, F., Poler, R., 2019. Low-code as enabler of digital transformation in manufacturing industry. Appl. Sci. 10 (1), 12. http://dx.doi.org/10.3390/app10010012.

Satell, G., 2017. Mapping Innovation: A Playbook for Navigating a Disruptive Age. McGraw-Hill Education New York, NY.

Schmidt, D.C., 2006. Guest editor's introduction: Model-driven engineering. Computer 39 (2), 25–31. http://dx.doi.org/10.1109/MC.2006.58.

Shull, F., Singer, J., Sjøberg, D.I., 2007. Guide to Advanced Empirical Software Engineering. Springer.

Simons, C., Singer, J., White, D.R., 2015. Search-based refactoring: Metrics are not enough. In: Search-Based Software Engineering - 7th International Symposium, SSBSE 2015, Bergamo, Italy, 2015, Proceedings. In: LNCS, vol. 9275, Springer, pp. 47–61. http://dx.doi.org/10.1007/978-3-319-22183-0_4.
Stephan, M., 2019. Emerging concepts and trends in collaborative modeling: A survey. In: Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2019, Prague, Czech Republic, 2019. SciTePress, pp. 240–247. http://dx.doi.org/10.5220/0007255502420249.
Stevenson, J., Wood, M.I., 2018. Recognising object-oriented software design quality: a practitioner-based questionnaire survey. Softw. Qual. J. 26 (2), 321–365. http://dx.doi.org/10.1007/s11219-017-9364-8.
Sunyé, G., 2017. Model consistency for distributed collaborative modeling. In: Modelling Foundations and Applications - 13th European Conference, Marburg, Germany. In: Lecture Notes in Computer Science, vol. 10376, Springer, pp. 197–212. http://dx.doi.org/10.1007/978-3-319-61482-3_12.
Syriani, E., Bill, R., Wimmer, M., 2019. Domain-specific model distance measures. J. Object Technol. 18 (3), 3:1–19. http://dx.doi.org/10.5381/jot.2019.18.3.a3.
Syriani, E., Vangheluwe, H., Mannadiar, R., Hansen, C., Mierlo, S.V., Ergin, H., 2013. AToMPM: A web-based modeling environment. In: Joint Proceedings of the 16th International Conference on Model Driven Engineering Languages and Systems, Miami, FL, USA. In: CEUR Workshop Proceedings, vol. 1115, CEUR-WS.org, pp. 21–25.
Taxén, L., 2006. An integration centric approach for the coordination of distributed software development projects. Inf. Softw. Technol. 48 (9), 767–780. http://dx.doi.org/10.1016/j.infsof.2006.01.007.
Tendeloo, Y.V., Vangheluwe, H., 2018. Unifying model- and screen sharing. In: 27th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, Paris, France. IEEE, pp. 127–132. http://dx.doi.org/10.1109/WETICE.2018.00031.
Tröls, M.A., Mashkoor, A., Egyed, A., 2019. Multifaceted consistency checking of collaborative engineering artifacts. In: 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, Munich, Germany. IEEE, pp. 278–287. http://dx.doi.org/10.1109/MODELS-C.2019.00044.
Van Mierlo, S., 2018. A multi-paradigm modelling approach for engineering model debugging environments (Ph.D. thesis). University of Antwerp, Belgium.
Van Mierlo, S., Deantoni, J., Burgueño, L., Verbrugge, C., Vangheluwe, H., 2019. Towards sketching interfaces for multi-paradigm modeling. In: 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, MODELS Companion 2019. IEEE, pp. 437–442. http://dx.doi.org/10.1109/MODELS-C.2019.00070.
Vaquero-Melchor, D., Palomares, J., Guerra, E., de Lara, J., 2017. Active domain-specific languages: Making every mobile user a modeller. In: 20th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Austin, TX, USA. IEEE, pp. 75–82. http://dx.doi.org/10.1109/MODELS.2017.13.
Vesin, B., Jolak, R., Chaudron, M.R.V., 2017. OctoUML: an environment for exploratory and collaborative software design. In: Uchitel, S., Orso, A., Robillard, M.P. (Eds.), Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, Companion Volume. IEEE, pp. 7–10. http://dx.doi.org/10.1109/ICSE-C.2017.19.
Visser, E., 2007. WebDSL: A case study in domain-specific language engineering. In: Generative and Transformational Techniques in Software Engineering II, International Summer School, Braga, Portugal. Revised Papers. In: LNCS, In: LNCS, vol. 5235, 291–373.http://dx.doi.org/10.1007/978-3-540-88643-3_7,
Vogels, W., 2009. Eventually consistent. Commun. ACM 52 (1), 40–44. http://dx.doi.org/10.1145/1435417.1435432.
Völter, M., Siegmund, J., Berger, T., Kolb, B., 2014. Towards user-friendly projectional editors. In: Software Language Engineering - 7th International Conference, VäSteråS, Sweden. Proceedings. In: LNCS, vol. 8706, Springer, pp. 41–61. http://dx.doi.org/10.1007/978-3-319-11245-9_3.
Voogd, S.N., Aslam, K., van Gool, L., Theelen, B., Malavolta, I., 2021. Real-time collaborative modeling across language workbenches - a case on jetbrains MPS and eclipse spoofax. In: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, Fukuoka, Japan. IEEE, pp. 16–26. http://dx.doi.org/10.1109/MODELS-C53483.2021.00011.
Weyssow, M., Sahraoui, H.A., Syriani, E., 2022. Recommending metamodel concepts during modeling activities with pre-trained language models. Softw. Syst. Model. 21 (3), 1071–1089. http://dx.doi.org/10.1007/s10270-022-00975-5.
Whitehead, J., 2007. Collaboration in software engineering: A roadmap. In: International Conference on Software Engineering, ISCE 2007, Workshop on the Future of Software Engineering, Minneapolis, MN, USA. IEEE, pp. 214–225. http://dx.doi.org/10.1109/FOSE.2007.4.
Wieland, K., Langer, P., Seidl, M., Wimmer, M., Kappel, G., 2013. Turning conflicts into collaboration. Comput. Support. Cooperative Work. 22 (2–3), 181–240. http://dx.doi.org/10.1007/s10606-012-9172-4.
Williams, L.A., 2001. Integrating pair programming into a software development process. In: 14th Conference on Software Engineering Education and Training, Charlotte, North Carolina, USA. IEEE, p. 27. http://dx.doi.org/10.1109/CSEE.2001.913816.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., 2012. Experimentation in Software Engineering. Springer, ISBN: 978-3-642-29043-5, http://dx.doi.org/10.1007/978-3-642-29044-2.
Wüest, D., Seyff, N., Glinz, M., 2019. FlexiSketch: a lightweight sketching and metamodeling approach for end-users. Softw. Syst. Model. 18 (2), 1513–1541. http://dx.doi.org/10.1007/s10270-017-0623-8.
Zadahmad, M., Syriani, E., Alam, O., Guerra, E., de Lara, J., 2019. Domain-specific model differencing in visual concrete syntax. In: Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering, Athens, Greece. ACM, pp. 100–112. http://dx.doi.org/10.1145/3357766.3359537.
Zhang, C., Budgen, D., 2013. A survey of experienced user perceptions about software design patterns. Inf. Softw. Technol. 55 (5), 822–835. http://dx.doi.org/10.1016/j.infsof.2012.11.003.
Zhang, Y., Patel, S., 2011. Agile model-driven development in practice. IEEE Softw. 28 (2), 84–91. http://dx.doi.org/10.1109/MS.2010.85.
Zhu, N., Grundy, J.C., Hosking, J.G., Liu, N., Cao, S., Mehra, A., 2007. Pounamu: A meta-tool for exploratory domain-specific visual language tool development. J. Syst. Softw. 80 (8), 1390–1407. http://dx.doi.org/10.1016/j.jss.2006.10.028.

Istvan David is a postdoctoral researcher at the University of Montréal, Canada. He received his Ph.D. in Computer Science from the University of Antwerp, Belgium. His research focuses on modeling-driven engineering, simulation, digital twins, and the automated construction of complex heterogeneous systems. He is also active outside of academia, mainly in innovation consulting.

Kousar Aslam is a postdoctoral researcher at the Vrije Universiteit Amsterdam, The Netherlands. She received her Ph.D. in Computer Science from the Eindhoven University of Technology. Her research focuses on blended modeling, software evolution and model-based re-engineering.

Ivano Malavolta is an associate professor at the Computer Science Department and Director of the Network Institute at the Vrije Universiteit Amsterdam (The Netherlands). His research focuses on data-driven software engineering, with a special emphasis on software architecture, mobile software development, robotics software. He applies empirical methods to evaluate methods and techniques in the field of software engineering. He authored several scientific articles in international journals and peer-reviewed international conferences proceedings. He is a member of ACM, IEEE, VERSEN, Amsterdam Young Academy, and Amsterdam Data Science.

Patricia Lago is a full professor in software engineering at Vrije Universiteit Amsterdam, The Netherlands, where she leads the Software and Sustainability research group in the Computer Science Department. Her passion in research is to create software engineering knowledge that makes software better, smarter, and more sustainable. Her research focuses on software architecture, software quality assessment, and software sustainability.