

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

^aDepartment of Computer Science and Operations Research, Université de Montréal, Canada

^bDepartment of Computer Science, Vrije Universiteit Amsterdam, The Netherlands

Abstract

The engineering of nowadays' 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, the available systematic studies on collaborative MDSE focus exclusively on mapping academic research and fail to identify how academic research, and industry practices and needs align. 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 have 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.

Keywords: Model-Driven Engineering, Collaborative Software Engineering, Industry Survey

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* [1, 2] 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 [3]. *Model-driven software engineering (MDSE)* answers these needs by addressing the inability of programming languages to alleviate the complexity of platforms and express do-

main concepts effectively [4]. 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 [5]. Collaborative MDSE has become a prominent feature of nowadays' software engineering practice [6], e.g., in agile methodologies and low-code platforms [7, 8, 9]. The rapid expansion and maturation of the field started attracting tool builders from outside of academia as well [10, 11, 12]. 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

*Corresponding author

Email address: istvan.david@umontreal.ca (Istvan David)

mapping and classifying academic research [13, 14]. 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.

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 [11]; and (ii) on the known systematic studies on the academic research on collaborative MDSE [14, 13, 15]. 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 [4, 6], 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.¹

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 toward 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 [16]. With its increasing adoption in practical settings [17], scaling MDSE to large, hierarchical teams has become a pressing need [18], 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).² 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 [19] and semantic [20] model differencing, model-based difference visualization [21], semantic inconsistency management [22], and intelligent editing assistance [23]. We aim to map which of these techniques are frequently encountered; and which ones are sought after by practitioners.

¹<https://github.com/david-istvan/collaborative-mdse-in-practice-replication-package>

²<https://www.omg.org/mof/>

According to our previous studies [13, 14] 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 [24], focus groups [25], and survey design [26]. 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.

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 toward 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 Figure 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 [27], and has been successfully adopted in software engineering research [28]. 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 Figure 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 [24]. Specifically, we directly contacted all industry partners within the consortium of the BUMBLE ITEA3 project [11], a European research project centered on blended modeling [29, 30] and collaborative modeling [13]. 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.”
- “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. [25]. 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 [25], the sessions lasted 90 minutes and were organized into four phases.

1. *Introduction* (5 minutes). A round of introduction among the participants and the moderator

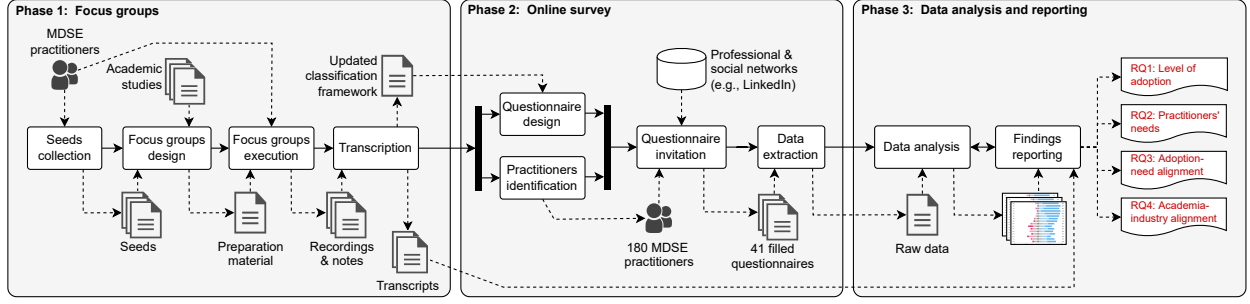


Figure 1: Overview of the study.

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).

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 minutes). 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 minutes). This phase is organized by the three dimensions of collaborative MDSE (see Section 2), enriched with (i) additional systematic *Academic studies* on collaborative MDSE [13, 14, 15] 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 minutes). 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 [31], 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 [13, 14] and the feature model of collaborative modeling defined in [15]. The resulting classification framework is shown in Table 2.

In addition to low-level updates of various fea-

Table 2: Classification framework for Collaborative MDSE

MODEL MANAGEMENT (29)		395	
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		400	
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		405	
Metrics of model complexity			
Natural Language Processing (for model building)			
Editors and modeling environments			
Visual editors		410	
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		415	
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		420	
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		425	
Eventual consistency			
Push notifications on conflicts			
Network architecture & robustness			
P2P (serverless) network architecture			
Cloud-based network architecture			
Failure recovery			
COMMUNICATION (25)			
Synchronous communication			
Chat		430	
Audio			
Voice			
Hand gestures			
Face-to-face			
Change review sessions			
Screen sharing			
Asynchronous communication			
Email		435	
Wiki			
Forum			
Proposals			
Voting			
Annotations			
Comments			
Feedback			
Reviews			
Call-For-Attention			
Sticky notes		440	
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			

tures 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 [13, 14, 15]. 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 Figure 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 [32, 33]. 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 Forms³ and is organized into seven sections.

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-pager⁴ for giving the definitions of

³<https://forms.google.com>

⁴Available in the replication package of the study – see Section 3.3

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 [34]. Likert-type scales have been used in practitioner survey in related domains, such as object-oriented software quality [35], search-based refactoring [36], and software design patterns [37]. Here, we measure the attitude of participants towards statements about the adoption of and 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) [38] and the International Hands-on Workshop on Collaborative Modeling (HoWCoM, 1 edition in 2021) [39].

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* [26]).

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 network⁵ and the Model-Driven Development Forum⁶), 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. Figure 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).

⁵<https://mde-network.com>

⁶<https://www.linkedin.com/groups/155446/>

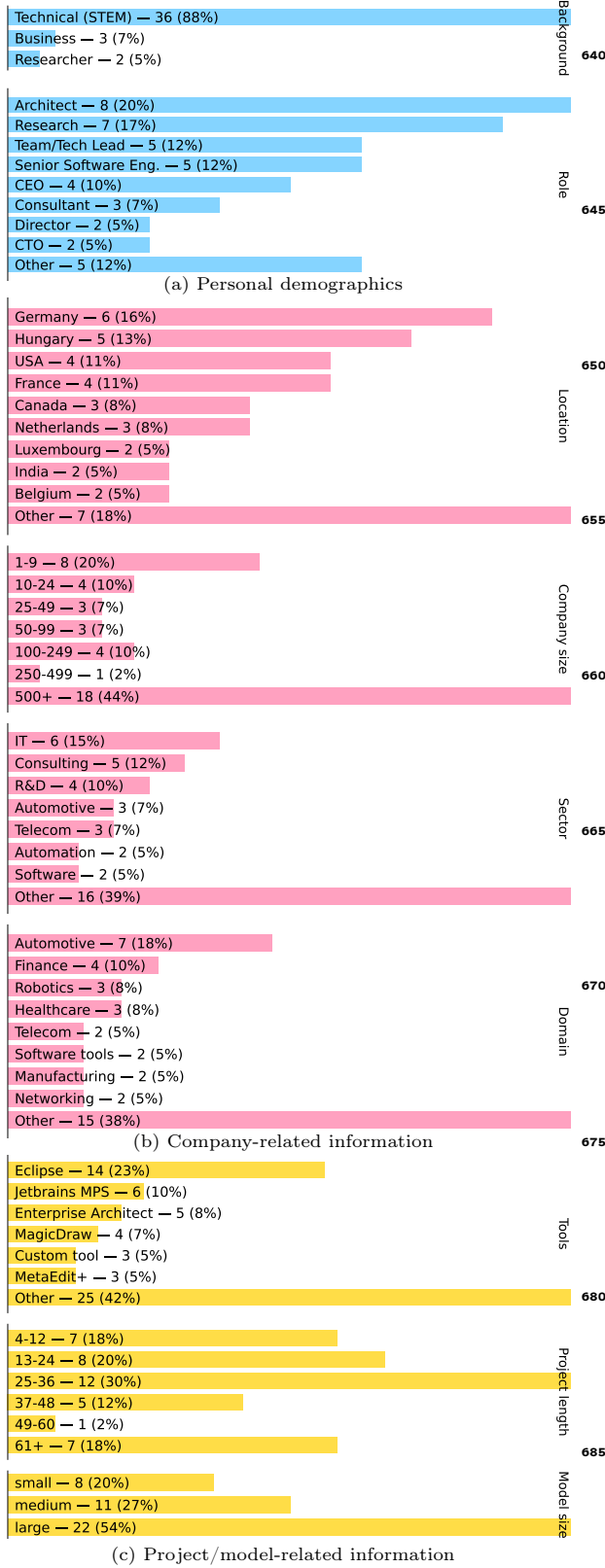


Figure 2: Demographics of survey participants

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 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 [34, 40, 41], and interpret "don't know" responses by ignoring them [42]. 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) [34, 41], 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 framework⁷ 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 [43]. The narrative synthesis method is widely used in secondary studies [44], 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

⁷<https://www.r-project.org/>

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 package¹ 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 Figures 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 package¹.

4.1. Model management

Figure 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%).

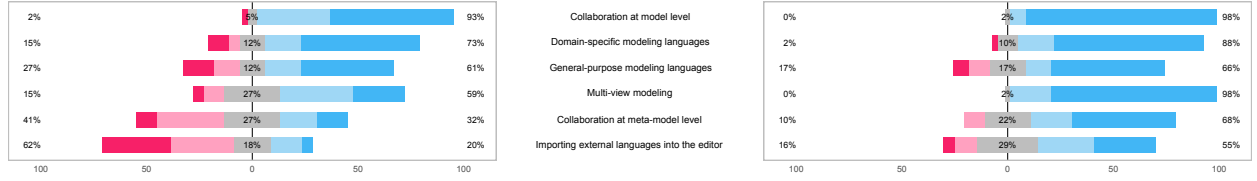
Model manipulation and query. *Model validation* (78%), *Model browsing/search* (76%) and *Code generation* (73%) are the three most adopted techniques of the group; and they score as the top three most needed features as well. Only these features score above 90% in terms of need in future projects. *Model transformations*, *Model debugging* [45] and *Model execution* also score high (above 80%) in the needs. Despite the recent improvements on the topic, *Natural Language Processing* [46, 47] is rarely used (8%). A similarly rare technique is establishing explicit *Metrics of model complexity* [48], 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* [49] 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%).

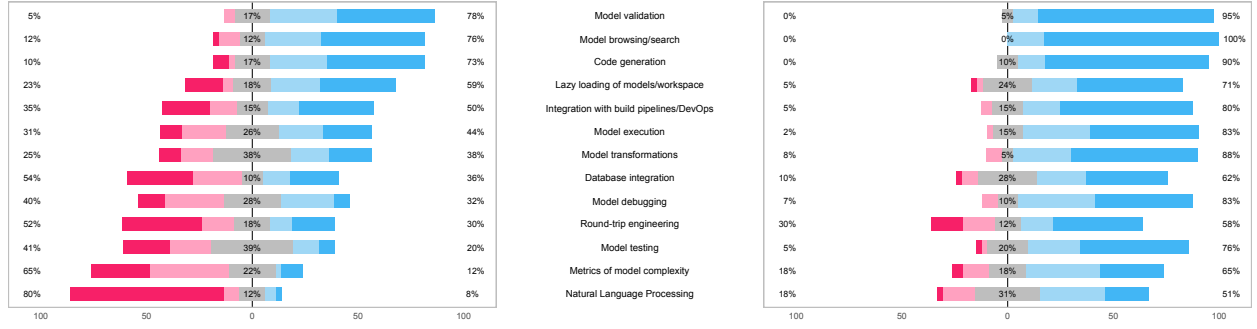
4.2. Collaboration

Figure 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.

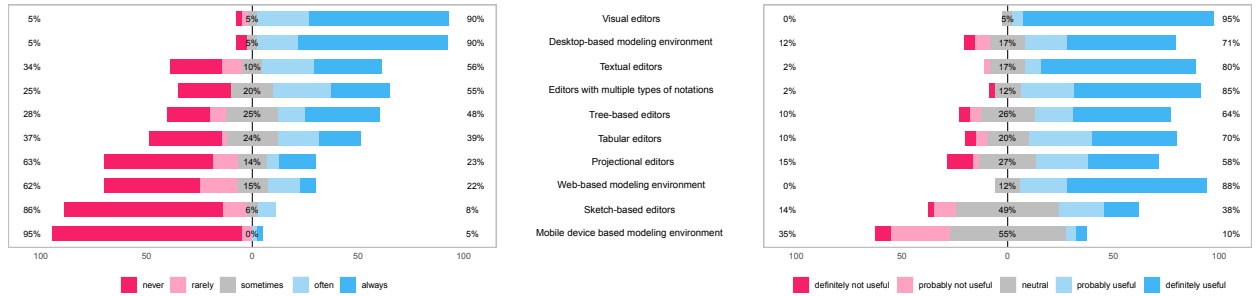
Stakeholder management and access control. 87% of the participants have used *User identification* techniques in their collaborative modeling work. *Authentication and authorization from corporate databases* (e.g., LDAP, AD) and *Role-based access control (RBAC)* are widely used as well at 71% and 69%,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



(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.

Figure 3: Adoption and needs in the *Model management* dimension.

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 [50].

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 tradition-

ally achieved by the means of version control systems such as Git and SVN, and MDSE adopted such techniques early on [51]. 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 [52], 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 reso-*

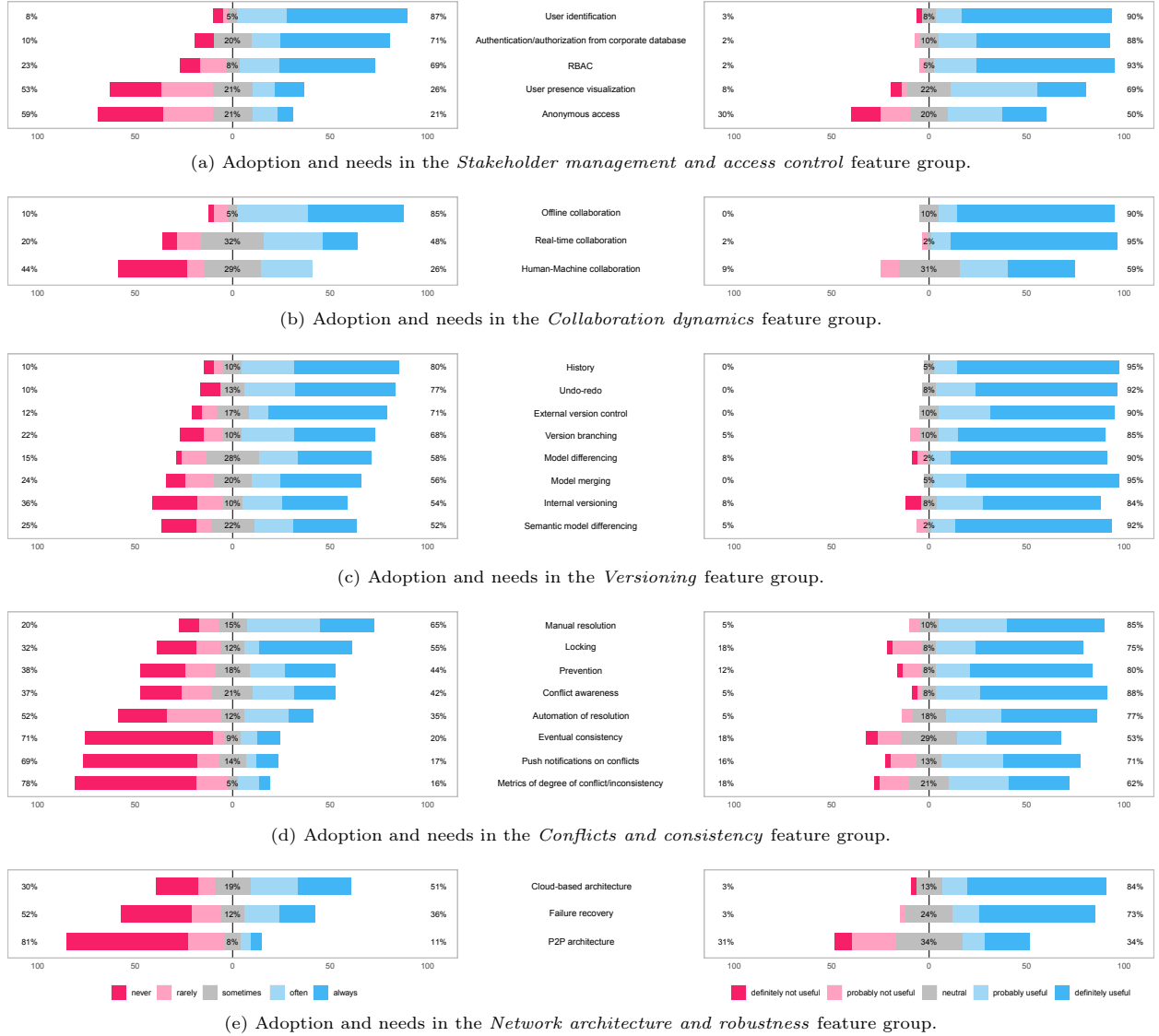


Figure 4: Adoption and needs in the *Collaboration* dimension.

lution 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 [53] (20%) and Metrics of inconsistency [54, 55] (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% 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%).

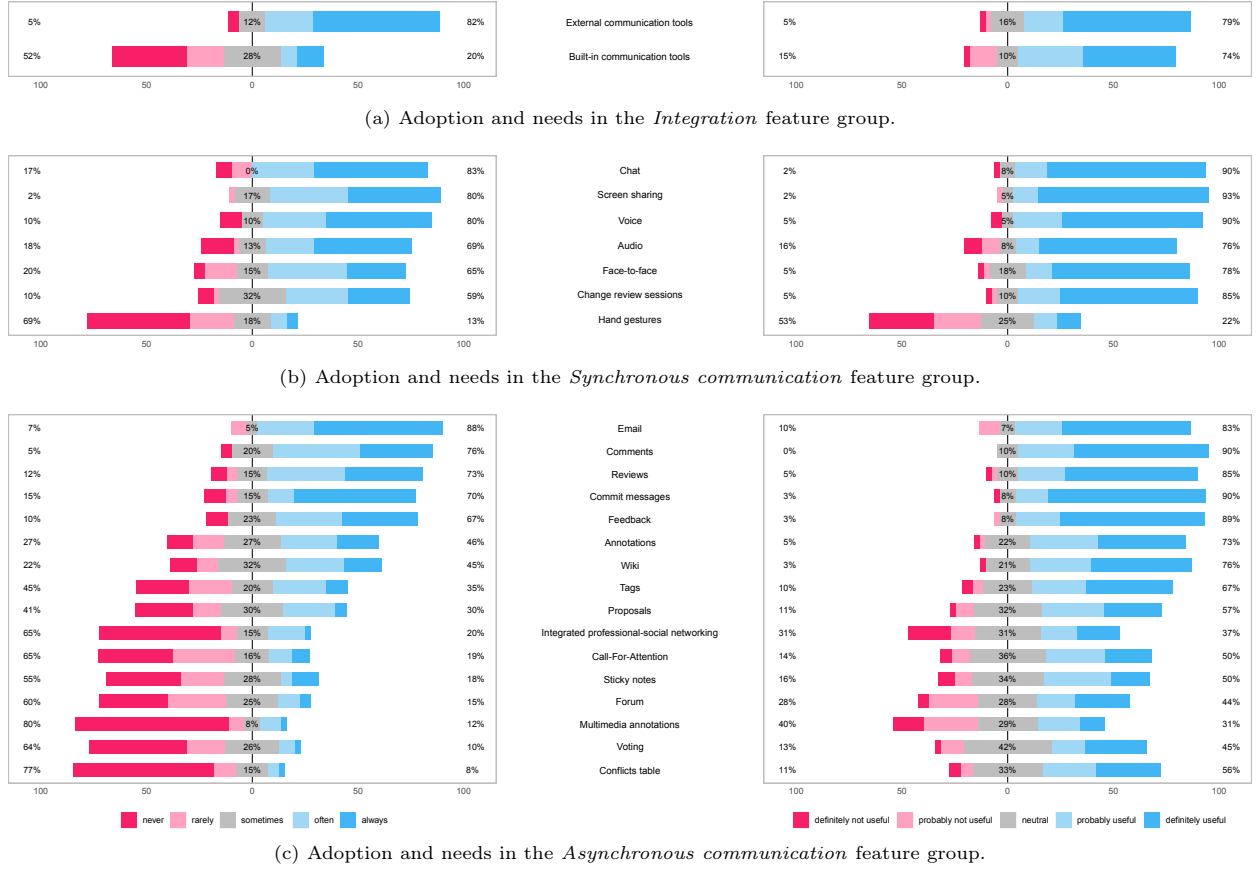


Figure 5: Adoption and needs in the *Communication* dimension.

4.3. Communication

Figure 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 devopment 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* [56] (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 5.1 and (Sections 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

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

(a) Adoption			(b) Need		
Feature	Feature group	%	Feature	Feature group	%
Collaboration at model level	Models and languages	93	Model browsing/search	Model manipulation and query	100
Visual editors	Editors and modeling environments	90	Multi-view modeling	Models and languages	98
Desktop-based environment	Editors and modeling environments	90	Collaboration at model level	Models and languages	98
Email	Asynchronous communication	88	Visual editors	Editors and modeling environments	95
User identification	Stakeholder management	87	Model validation	Model manipulation and query	95
Offline collaboration	Collaboration dynamics	85	History	Versioning	95
Chat	Synchronous communication	83	Real-time collaboration	Collaboration dynamics	95
External communication tools	Integration	82	Model merging	Versioning	95
History	Versioning	80	Role-based access control	Stakeholder management	93
Screen sharing	Synchronous communication	80	Screen sharing	Synchronous communication	93
Forum		
Hand gestures	Asynchronous communication	15	Call-For-Attention	Asynchronous communication	50
Metrics of model complexity	Synchronous communication	13	Anonymous access	Stakeholder management	50
Multimedia annotations	Model manipulation and query	12	Voting	Asynchronous communication	45
Peer-to-peer architecture	Asynchronous communication	12	Forum	Asynchronous communication	44
Voting	Network architecture and robustness	11	Sketch-based editors	Editors and modeling environments	38
Sketch-based editors	Asynchronous communication	10	Integrated prof.-soc. networking	Asynchronous communication	37
Conflicts table	Editors and modeling environments	8	Peer-to-peer architecture	Network architecture and robustness	34
Natural Language Processing	Asynchronous communication	8	Multimedia annotations	Asynchronous communication	31
Mobile environment	Model manipulation and query	8	Hand gestures	Synchronous communication	22
	Editors and modeling environments	5	Mobile environment	Editors and modeling environments	10

al. [57] and Akdur et al.[58] 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 [59] (6; 10%), Enterprise Architect [60] (5, 8%), MagicDraw [61], and MetaEdit+ [62] (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 [63], suggesting 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 Figure 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 [64], 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*.

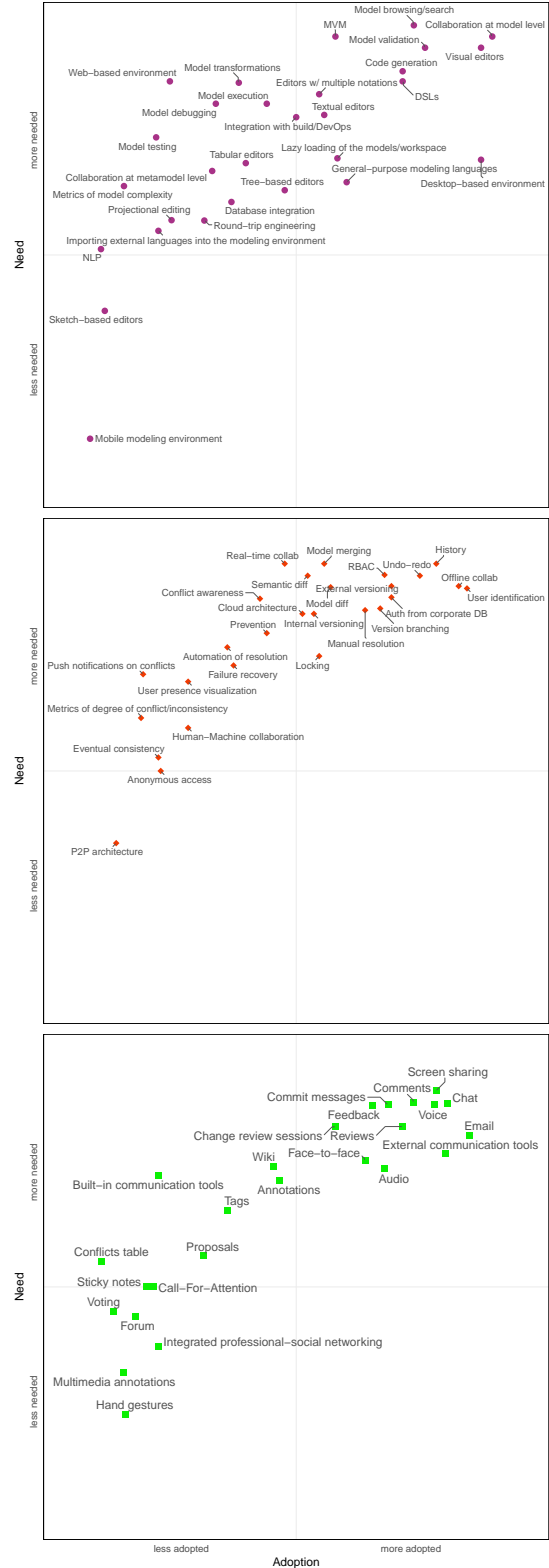


Figure 6: Need-Adoption Matrix of the three dimensions: ● Model management, ◆ Collaboration, ■ Communication.

In the following we provide the main insights emerging from the Need-adoption matrix⁸. 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 [65].

Some characteristic examples include *Sketch-based editors* [66] and *Mobile modeling environments* [67], 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 *breakout* 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 [65].

The characteristic examples of this quadrant outline a plausible concept of the next generation of collaborative modeling tools, implemented in *Web-based environments* [68], with a combination of *Real-time* [69, 70] collaboration capabilities possibly driven by *Eventual consistency models* [53], and augmented by *Built-in communication tools*, with the added capabilities of *Model debugging* [45] and *Model testing* [71]. 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’s worth an effort to do something there – P2*”.

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 [65], suggesting incremental improvements to existing features. We remark, however, that features of this quadrant also provide opportunities for

⁸Our sample does not record collaborative MDSE features in the *Less needed – More Adopted* quadrant, thus, we do not discuss it.

1100 *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.

1120 The characteristic examples of this quadrant outline a typical state-of-the-practice collaborative modeling tool, implemented in *Desktop environments* [62], predominantly equipped with *Visual editing* capabilities, supporting *Domain-specific modeling* [72]; 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* [73]; 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)

1135 To answer RQ4, we build on the systematic mapping studies by Franzago et al. [13] (covering research output until 2013) and David et al. [14] (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 [13] and [14]. To allow the mapping, two researchers identified the overlap between the features of the current work (Table 2) and previous works [13, 14]. A third researcher validated the results. Because [13] and [14] 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

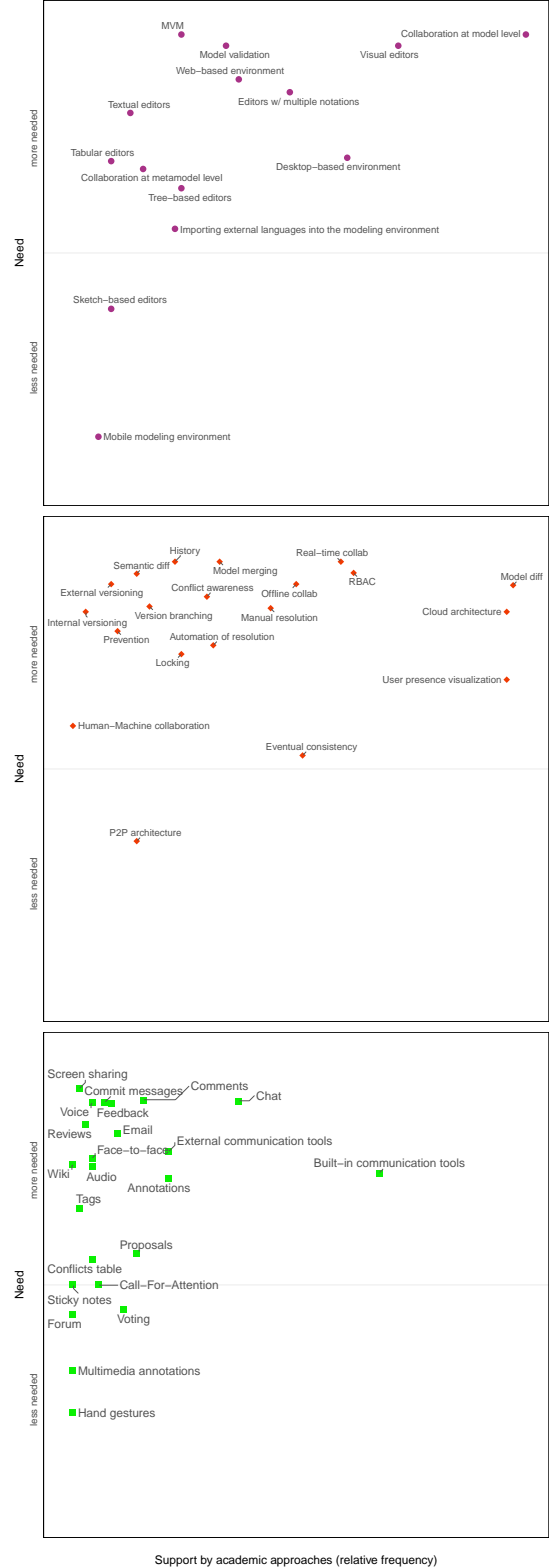


Figure 7: Needs-Academia Matrix of the three dimensions: ● Model management, ◆ Collaboration, ■ Communication.

focus group discussions and could not be mapped directly. For the sake of replicability and independent verification, our replication package¹ 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 Figure 7. Similarly to Section 5.3, the charts provide visual support with *relative* values. For the correct interpretation of the matrices, 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 [65].

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 [13] and [14] 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 [13].

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 [73] and [74], model validation via OCL constraints in [75] and via a critics-based approach in [76], and web-based environments in WebGME [77] and CrowdUI [78]. 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 [79], such as Xtext for the Eclipse Modeling Framework, but not in the context of collaborative MDSE [80, 81].

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 [66] and OctoUML [82] are some of the few sketch-based modeling tools published since 2016 [14]. Sketch-based editors have been researched outside of the collaborative MDSE domain, e.g., by Van Mierlo et al.[83]. The situation is different in *Mobile modeling environments*, where the number of academic approaches increased to four between 2016–2020 [14], while there were none before 2016 [13]. 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 dataset of [13] and [14]. Examples of academic approaches studying external version control systems in the context of collaborative MDSE are [84, 85, 86]. Collaboration by internal model-based version control has been demonstrated in prototype modeling tools as well [52, 87]. It is important to note that advanced versioning techniques, such as domain-specific [19] and semantic model differencing [88] that leverage the semantically rich context of MDSE are trending topics in academic research [14]. 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 [14, 13], aligning well with its 95% industry need. Especially driven by relaxed consistency models (such as eventual consistency [53] and strong eventual consistency [89]), the feasibility of real-time collaborative modeling has been demonstrated in conceptual modeling [90], requirements modeling [91], cross-platform collaborative modeling [92], and collaborative multi-level modeling [69]. A combination of *Real-time collaboration* and traditional *Versioning* (Git) has been recently implemented in WebGME [52].

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 emer-

gence of Web-based IDEs such as Eclipse Theia⁹, WebMPS¹⁰, 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 [90], performance improvement by direct model change messages [93], and flexibility in terms of dynamically joining collaborators [94].

5.4.3. Communication

As evidenced by Figure 7, industry needs in the Communication dimension are in striking contrast with academic research. As concluded by the academic studies [13, 14], 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 [95, 96, 52]. Unified model- and screen sharing has been researched as well [74], 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)¹¹ and the International Conference on Global Software Engineering (ICGSE)¹². 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 (COMMit-MDE, 3 editions between 2016–2018) [38] and the International Hands-on Workshop on Collaborative Modeling (HoWCoM, 2021) [39].

In this context, *pair modeling* might be an interesting topic to investigate. We envision pair modeling following principles similar to those of pair

⁹<https://theia-ide.org>

¹⁰<https://blog.jetbrains.com/mps/2021/02/webmps-faq>

¹¹<https://conf.researchr.org/home/chase-2021>

¹²<https://conf.researchr.org/series/icgse>

programming in traditional software engineering—defined as two developers working side-by-side at one computer, collaborating on the same code [97], 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 [98]. Remote pair programming has been gaining particular popularity recently, as demonstrated by the targeted features of mainstream programming environments, such as Code With Me¹³ by JetBrains and Teletype for Atom¹⁴. 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 [13, 14, 15]. 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. [99] 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 [13, 14] and feature models [15], 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 [40]. Standalone and independent Likert items, such as the ones in our study, can be safely analyzed by modes, medians, and frequencies [41]. Due to the large number of elements in our framework (Table 2) not allowing for a concise discussion of the results in these terms, 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

¹³<https://www.jetbrains.com/code-with-me/>

¹⁴<https://teletype.atom.io/>

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); 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 [100, 101, 102]. 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 [103]. Further exploration of the factors affecting effective collaboration was studied through questionnaire-guided interviews in globally distributed software development projects [104]. Collaboration during specific phases of software development life cycle, such as software design, is studied through a set of interviews conducted with software architects working at a large global software solutions provider [105]. 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. [106] 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. [105]. 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. [107] 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 [58, 108, 109].

Surveys on collaborative MDSE. The state of the art in collaborative MDSE has been systematically assessed in previous studies [13, 14]. 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. [15] on the features of collaborative modeling tools; and Stephan [110] 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. [111].

Acknowledgements

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

- [1] J. D. Herbsleb, 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, 2007, pp. 188–198. doi:10.1109/FOSE.2007.11.
- [2] J. Whitehead, 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, 2007, pp. 214–225. doi:10.1109/FOSE.2007.4.
- [3] M. Persson, M. Törngren, A. Qamar, J. Westman, M. Biehl, S. Tripakis, H. Vangheluwe, J. Denil, 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, 2013, pp. 10:1–10:10. doi:10.1109/EMSOF.2013.6658588.
- [4] D. C. Schmidt, Guest editor's introduction: Model-driven engineering, *Computer* 39 (2) (2006) 25–31. doi:10.1109/MC.2006.58.
- [5] H. Muccini, J. Bosch, A. van der Hoek, Collaborative modeling in software engineering, *IEEE Softw.* 35 (6) (2018) 20–24. doi:10.1109/MS.2018.4321244.
- [6] M. Brambilla, J. Cabot, M. Wimmer, *Model-driven software engineering in practice*, second edition (2017). doi:10.2200/S00751ED2V01Y201701SWE004.
- [7] Y. Zhang, S. Patel, Agile model-driven development in practice, *IEEE Softw.* 28 (2) (2011) 84–91. doi:10.1109/MS.2010.85.
- [8] R. Sanchis, Ó. García-Perales, F. Fraile, R. Poler, Low-code as enabler of digital transformation in manufacturing industry, *Applied Sciences* 10 (1) (2019) 12. doi:10.3390/app10010012.
- [9] J. Cabot, 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, 2020, pp. 76:1–76:3. doi:10.1145/3417990.3420210.
- [10] MONDO Project official webpage, <http://www.mondo-project.org/>, Retrieved: 14 Oct, 2022.
- [11] BUMBLE Project official webpage, <https://itea3.org/project/bumble.html>, Retrieved: 14 Oct, 2022.
- [12] Lowcomote official webpage, <https://www.lowcomote.eu>, Retrieved: 14 Oct, 2022.
- [13] M. Franzago, D. D. Ruscio, I. Malavolta, H. Muccini, Collaborative model-driven software engineering: A classification framework and a research map, *IEEE Trans. Software Eng.* 44 (12) (2018) 1146–1175. doi:10.1109/TSE.2017.2755039.
- [14] I. David, K. Aslam, S. Faridmoayer, I. Malavolta, E. Syriani, P. Lago, 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, 2021, pp. 273–284. doi:10.1109/MODELS50736.2021.00035.
- [15] C. Masson, J. Corley, E. Syriani, Feature model for collaborative modeling environments, in: *Proceedings of MODELS 2017 Satellite Events*, Austin, TX, USA, September, 17, 2017, Vol. 2019 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 164–173.
- [16] J. Y. Bang, Y. Brun, N. Medvidovic, Continuous analysis of collaborative design, in: *2017 IEEE International Conference on Software Architecture*, Gothenburg, Sweden, 2017, IEEE, 2017, pp. 97–106. doi:10.1109/ICSA.2017.45.
- [17] A. Bucchiarone, F. Ciccozzi, L. Lambers, A. Pierantonio, M. Tichy, M. Tisi, A. Wortmann, V. Zaytsev, What is the future of modeling?, *IEEE Softw.* 38 (2) (2021) 119–127. doi:10.1109/MS.2020.3041522.
- [18] D. S. Kolovos, L. M. Rose, N. D. Matragkas, R. F. Paige, E. Guerra, J. S. Cuadrado, J. de Lara, I. Ráth, D. Varró, M. Tisi, J. Cabot, A research roadmap towards achieving scalability in model driven engineering, in: *Proceedings of the Workshop on Scalability in Model Driven Engineering*, Budapest, Hungary, ACM, 2013, p. 2. doi:10.1145/2487766.2487768.
- [19] M. Zadahmad, E. Syriani, O. Alam, E. Guerra, J. de Lara, Domain-specific model differencing in visual concrete syntax, in: *Proceedings of the 12th ACM*

- SIGPLAN International Conference on Software Lan-1725
guage Engineering, Athens, Greece, ACM, 2019, pp.
100–112. doi:10.1145/3357766.3359537.
- [20] T. Kehrer, U. Kelter, G. Taentzer, A rule-based ap-1730
proach to the semantic lifting of model differences in
the context of model versioning, in: 26th IEEE/ACM 1730
International Conference on Automated Software En-
gineering, Lawrence, KS, USA, IEEE, 2011, pp. 163–
172. doi:10.1109/ASE.2011.6100050.
- [21] P. Brosch, M. Seidl, M. Wimmer, G. Kappel, Con-1735
flict visualization for evolving UML models, J. Object
Technol. 11 (3) (2012) 2: 1–30. doi:10.5381/jot. 1735
2012.11.3.a2.
- [22] I. David, J. Denil, K. Gadeyne, H. Vangheluwe,
Engineering process transformation to manage
(in)consistency, in: Proceedings of the 1st Inter-1740
national Workshop on Collaborative Modelling in
MDE co-located with ACM/IEEE 19th International
Conference on Model Driven Engineering Languages
and Systems, St. Malo, France, Vol. 1717 of CEUR
Workshop Proceedings, CEUR-WS.org, 2016, pp. 1745
7–16.
- [23] G. Mussbacher, et al., Opportunities in intelligent
modeling assistance, Softw. Syst. Model. 19 (5) (2020)
1045–1053. doi:10.1007/s10270-020-00814-5.
- [24] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, 1750
B. Regnell, Experimentation in Software Engineering,
Springer, 2012. doi:10.1007/978-3-642-29044-2.
- [25] J. Kontio, J. Bragge, L. Lehtola, The focus group
method as an empirical tool in software engineer-
ing, in: Guide to Advanced Empirical Software En-1755
gineering, Springer London, 2008, pp. 93–116. doi:
10.1007/978-1-84800-044-5_4.
- [26] B. A. Kitchenham, S. L. Pfleeger, Principles of sur-
vey research: part 5: populations and samples, ACM
SIGSOFT Softw. Eng. Notes 27 (5) (2002) 17–20. 1760
doi:10.1145/571681.571686.
- [27] S. B. Merriam, E. J. Tisdell, Qualitative research: A
guide to design and implementation, John Wiley &
Sons, 2015.
- [28] J. Kontio, L. Lehtola, J. Bragge, Using the focus group 1765
method in software engineering: Obtaining practi-
tioner and user experiences, in: 2004 International
Symposium on Empirical Software Engineering (IS-
ESE 2004), 2004, Redondo Beach, CA, USA, IEEE,
2004, pp. 271–280. doi:10.1109/ISESE.2004.35. 1770
- [29] I. David, M. Latifaj, J. Pietron, W. Zhang, F. Ci-
ccozzi, I. Malavolta, A. Raschke, J.-P. Steghöfer,
R. Hebig, Blended modeling in commercial and open-
source model-driven software engineering tools: A
systematic study, Softw. Syst. Model. (2022). doi: 1775
10.1007/s10270-022-01010-3.
- [30] M. Latifaj, F. Ciccozzi, M. W. Anwar, M. Mohlin,
Blended graphical and textual modelling of UML-RT
state-machines: An industrial experience, in: Software
Architecture - 15th European Conference, Tracks and 1780
Workshops; Växjö, Sweden, Revised Selected Papers,
Vol. 13365 of LNCS, Springer, 2021, pp. 22–44. doi:
10.1007/978-3-031-15116-3_2.
- [31] D. G. Oliver, J. M. Serovich, T. L. Mason, Con-
straints and opportunities with interview transcrip-
tion: Towards reflection in qualitative research, Social
Forces 84 (2) (2005) 1273–1289. doi:10.1353/sof. 1785
2006.0023.
- [32] F. Shull, J. Singer, D. I. Sjöberg, Guide to advanced
empirical software engineering, Springer, 2007.
- [33] A. N. Oppenheim, Questionnaire design, interviewing
and attitude measurement, Bloomsbury Publishing,
2000.
- [34] A. Joshi, et al., Likert scale: Explored and explained,
British Journal of Applied Science & Technology 7 (4)
(2015) 396.
- [35] J. Stevenson, M. I. Wood, Recognising object-oriented
software design quality: a practitioner-based question-
naire survey, Softw. Qual. J. 26 (2) (2018) 321–365.
doi:10.1007/s11219-017-9364-8.
- [36] C. Simons, J. Singer, D. R. White, Search-based refac-
toring: Metrics are not enough, in: Search-Based Soft-
ware Engineering - 7th International Symposium, SS-
BSE 2015, Bergamo, Italy, 2015, Proceedings, Vol.
9275 of LNCS, Springer, 2015, pp. 47–61. doi:10.
1007/978-3-319-22183-0_4.
- [37] C. Zhang, D. Budgen, A survey of experienced user
perceptions about software design patterns, Inf. Softw.
Technol. 55 (5) (2013) 822–835. doi:10.1016/j. 1790
infsof.2012.11.003.
- [38] J. Bosch, S. Gerard, D. S. Kolovos, I. Malavolta,
H. Muccini, Commitmde 2018 - 3rd international
workshop on collaborative modelling in MDE, in: Pro-
ceedings of MODELS 2018 Workshops co-located with
ACM/IEEE 21st International Conference on Model
Driven Engineering Languages and Systems, Copen-
hagen, Denmark, Vol. 2245 of CEUR Workshop Pro-
ceedings, CEUR-WS.org, 2018, pp. 266–267.
- [39] I. David, E. Syriani, A. García-Domínguez, Preface to
the 1st international hands-on workshop on collabora-
tive modeling (howcom 2021), in: ACM/IEEE Inter-
national Conference on Model Driven Engineering
Languages and Systems Companion, Fukuoka, Japan,
IEEE, 2021, pp. 1–2. doi:10.1109/MODELS-C53483.
2021.00001.
- [40] G. Albaum, The likert scale revisited, Market Research
Society. Journal. 39 (2) (1997) 1–21. doi:10.1177/
147078539703900202.
- [41] S. E. Harpe, How to analyze likert and other rating
scale data, Currents in Pharmacy Teaching and Learn-
ing 7 (6) (2015) 836–850. doi:10.1016/j.cptl.2015.
08.001.
- [42] S. Y. Y. Chyung, K. Roberts, I. Swanson, A. Hankin-
son, Evidence-based survey design: The use of a mid-
point on the likert scale, Performance Improvement
56 (10) (2017) 15–23. doi:10.1002/pfi.21727.
- [43] M. Rodgers, A. Sowden, M. Petticrew, L. Arai,
H. Roberts, N. Britten, J. Popay, Testing methodolog-
ical guidance on the conduct of narrative synthesis in
systematic reviews, Evaluation 15 (1) (2009) 49–73.
doi:10.1177/1356389008097871.
- [44] D. S. Cruzes, T. Dybå, Research synthesis in soft-
ware engineering: A tertiary study, Information and
Software Technology 53 (5) (2011) 440–455. doi:
10.1016/j.infsof.2011.01.004.
- [45] S. Van Mierlo, A multi-paradigm modelling approach
for engineering model debugging environments, Ph.D.
thesis, University of Antwerp, Belgium (2018).
- [46] M. Weyssow, H. A. Sahraoui, E. Syriani, Recom-
mending metamodel concepts during modeling activi-
ties with pre-trained language models, Softw. Syst.
Model. 21 (3) (2022) 1071–1089. doi:10.1007/
s10270-022-00975-5.
- [47] L. Burgueño, R. Clarisó, S. Gérard, S. Li, J. Cabot,

- 1790 An nlp-based architecture for the autocompletion
of partial domain models, in: *Advanced Informa- 1855*
tion Systems Engineering - 33rd International Con-
ference, Melbourne, VIC, Australia, Vol. 12751 of
LNCS, Springer, 2021, pp. 91–106. doi:10.1007/
1795 *978-3-030-79382-1_6. 1860*
- [48] G. Polancic, B. Cegnar, Complexity metrics for pro-
cess models - A systematic literature review, *Comput.*
Stand. Interfaces 51 (2017) 104–117. doi:10.1016/j.
csi.2016.12.003.
- 1800 [49] M. Völter, J. Siegmund, T. Berger, B. Kolb, To-
wards user-friendly projectional editors, in: *Soft- 1865*
ware Language Engineering - 7th International Con-
ference, Västerås, Sweden. Proceedings, Vol. 8706
of LNCS, Springer, 2014, pp. 41–61. doi:10.1007/
1805 *978-3-319-11245-9_3. 1870*
- [50] S. Pérez-Soler, E. Guerra, J. de Lara, Collaborative
modeling and group decision making using chatbots
in social networks, *IEEE Softw.* 35 (6) (2018) 48–54.
doi:10.1109/MS.2018.290101511.
- 1810 [51] A. D. Lucia, F. Fasano, G. Scanniello, G. Tortora,
Enhancing collaborative synchronous UML modelling
with fine-grained versioning of software artefacts, *J.*
Vis. Lang. Comput. 18 (5) (2007) 492–503. doi:10.
1016/j.jvlc.2007.08.005.
- 1815 [52] M. Maróti, R. Kereskényi, T. Kecskés, P. Völgyesi,
Á. Lédeczi, Online collaborative environment for de-
signing complex computational systems 29 (2014)
2432–2441. doi:10.1016/j.procs.2014.05.227.
- [53] W. Vogels, Eventually consistent, *Commun. ACM*
52 (1) (2009) 40–44. doi:10.1145/1435417.1435432.
- 1820 [54] E. Syriani, R. Bill, M. Wimmer, Domain-specific
model distance measures, *J. Object Technol.* 18 (3)
(2019) 3:1–19. doi:10.5381/jot.2019.18.3.a3.
- [55] I. David, E. Syriani, C. Verbrugge, D. Buchs,
D. Blouin, A. Cicchetti, K. Vanherpen, Towards in-
consistency tolerance by quantification of semantic in-
consistencies, in: *Proceedings of the 1st International*
Workshop on Collaborative Modelling in MDE co-
located with ACM/IEEE 19th International Confer-
ence on Model Driven Engineering Languages and Sys-
tems, St. Malo, France, Vol. 1717 of CEUR Workshop
Proceedings, CEUR-WS.org, 2016, pp. 35–44.
- 1830 [56] N. Boulila, Group support for distributed collabora-
tive concurrent software modeling, in: *19th IEEE In-*
ternational Conference on Automated Software En-
gineering, Linz, Austria, IEEE, 2004, pp. 422–425.
doi:10.1109/ASE.2004.10014.
- 1835 [57] J. E. Hutchinson, J. Whittle, M. Rouncefield, Model-
driven engineering practices in industry: Social, orga-
nizational and managerial factors that lead to success
or failure, *Sci. Comput. Program.* 89 (2014) 144–161.
doi:10.1016/j.scico.2013.03.017.
- 1840 [58] D. Akdur, V. Garousi, O. Demirörs, A survey on mod-
eling and model-driven engineering practices in the
embedded software industry, *J. Syst. Archit.* 91 (2018)
62–82. doi:10.1016/j.sysarc.2018.09.007.
- 1845 [59] V. Pech, A. Shatalin, M. Voelter, 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 Ma-
chines, Languages, and Tools, Stuttgart, Germany,
ACM, 2013, pp. 165–168. doi:10.1145/2500828.
2500846.
- 1850 [60] Enterprise Architect website, <https://sparxsystems.com/products/ea/index.html>, Retrieved: 14 Oct,
2022.
- [61] E. Planas, J. Cabot, How are UML class diagrams
built in practice? A usability study of two UML tools:
Magicdraw and papyrus, *Comput. Stand. Interfaces* 67
(2020). doi:10.1016/j.csi.2019.103363.
- [62] S. Kelly, J. Tolvanen, Collaborative creation and ver-
sioning 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, 2018, pp. 37–41. doi:10.1145/
3270112.3270132.
- [63] I. Mistrik, J. Grundy, A. van der Hoek, J. White-
head, Collaborative software engineering: Challenges
and prospects, in: *Collaborative Software Engineer-*
ing, Springer, 2010, pp. 389–403. doi:10.1007/
978-3-642-10294-3_19.
- [64] B. D. Henderson, The product portfolio, *The Boston*
Consulting Group Perspective 66 (1970).
- [65] G. Satell, Mapping innovation: a playbook for navi-
gating a disruptive age, McGraw-Hill Education New
York, NY, 2017.
- [66] D. Wüest, N. Seyff, M. Glinz, Flexisketch: a
lightweight sketching and metamodeling approach for
end-users, *Softw. Syst. Model.* 18 (2) (2019) 1513–
1541. doi:10.1007/s10270-017-0623-8.
- [67] D. Vaquero-Melchor, J. Palomares, E. Guerra,
J. de Lara, Active domain-specific languages: Making
every mobile user a modeller, in: *20th ACM/IEEE*
International Conference on Model Driven Engineer-
ing Languages and Systems, Austin, TX, USA, IEEE,
2017, pp. 75–82. doi:10.1109/MODELS.2017.13.
- [68] J. Corley, E. Syriani, H. Ergin, S. Van Mierlo, Cloud-
based multi-view modeling environments, in: *Modern*
Software Engineering Methodologies for Mobile and
Cloud Environments, IGI Global, 2016, pp. 120–139.
doi:10.4018/978-1-4666-9916-8.ch007.
- [69] I. David, E. Syriani, Real-time collaborative multi-
level modeling by conflict-free replicated data
types, *Softw. Syst. Model.* (2022). doi:10.1007/
s10270-022-01054-5.
- [70] I. David, E. Syriani, C. Masson, Extensible conflict-
free replicated datatypes for real-time collaborative
software engineering, in: *Proceedings of the 17th Con-*
ference on Computer Science and Intelligence Systems,
Sofia, Bulgaria, Vol. 30 of Annals of Computer Sci-
ence and Information Systems, 2022, pp. 849–853.
doi:10.15439/2022F99.
- [71] S. Nejati, K. Gaaloul, C. Menghi, L. C. Briand,
S. Foster, D. Wolfe, Evaluating model testing and
model checking for finding requirements violations in
simulink models, in: *Proceedings of the ACM Joint*
Meeting on European Software Engineering Confer-
ence and Symposium on the Foundations of Software
Engineering, Tallinn, Estonia, ACM, 2019, pp. 1015–
1025. doi:10.1145/3338906.3340444.
- [72] E. Visser, Webdsl: A case study in domain-specific
language engineering 5235 (2007) 291–373. doi:10.
1007/978-3-540-88643-3_7.
- [73] A. Gómez, X. Mendiadua, K. Barmpis, G. Bergmann,
J. Cabot, X. D. Carlos, C. Debrececi, A. Garmen-
dia, D. S. Kolovos, J. de Lara, Scalable model-
ing technologies in the wild: an experience report
on wind turbines control applications development,

- Softw. Syst. Model. 19 (5) (2020) 1229–1261. doi:10.1007/s10270-020-00776-8.
- [74] Y. V. Tendeloo, H. Vangheluwe, Unifying model- and screen sharing, in: 27th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, Paris, France, IEEE, 2018, pp. 127–132. doi:10.1109/WETICE.2018.00031.
- [75] N. Zhu, J. C. Grundy, J. G. Hosking, N. Liu, S. Cao, A. Mehra, Pounamu: A meta-tool for exploratory domain-specific visual language tool development, J. Syst. Softw. 80 (8) (2007) 1390–1407. doi:10.1016/j.jss.2006.10.028.
- [76] K. Wieland, P. Langer, M. Seidl, M. Wimmer, G. Kappel, Turning conflicts into collaboration, Comput. Support. Cooperative Work. 22 (2-3) (2013) 181–240. doi:10.1007/s10606-012-9172-4.
- [77] M. Maróti, T. Kecskés, R. Kereskényi, B. Broll, P. Völgyesi, L. Jurácz, T. Levendovszky, Á. Lédeczi, Next generation (meta) modeling: Web- and cloud-based collaborative tool infrastructure 1237 (2014) 41–60.
- [78] J. Oppenlaender, T. Tiropanis, S. Hosio, CrowdUI: Supporting web design with the crowd, Proc. ACM Hum. Comput. Interact. 4 (EICS) (2020) 76:1–76:28. doi:10.1145/3394978.
- [79] B. Mierle, Textual modeling tools: overview and comparison of language workbenches, in: W. R. Cook, S. Clarke, M. C. Rinard (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, 2010, pp. 139–148. doi:10.1145/1869542.1869564.
- [80] S. Erdweg, et al., 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, Vol. 8225 of LNCS, Springer, 2013, pp. 197–217. doi:10.1007/978-3-319-02654-1_11.
- [81] M. Eysholdt, H. Behrens, 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, 2010, pp. 307–309. doi:10.1145/1869542.1869625.
- [82] B. Vesin, R. Jolak, M. R. V. Chaudron, Octouml: an environment for exploratory and collaborative software design, in: S. Uchitel, A. Orso, M. P. Robillard (Eds.), Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, Companion Volume, IEEE, 2017, pp. 7–10. doi:10.1109/ICSE-C.2017.19.
- [83] S. Van Mierlo, J. Deantoni, L. Burgueño, C. Verbrugge, H. Vangheluwe, 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, 2019, pp. 437–442. doi:10.1109/MODELS-C.2019.00070.
- [84] J. D. Rocco, D. D. Ruscio, L. Iovino, A. Pierantonio, Collaborative repositories in model-driven engineering, IEEE Softw. 32 (3) (2015) 28–34. doi:10.1109/MS.2015.61.
- [85] D. Kuryazov, A. Winter, R. H. Reussner, Collaborative modeling enabled by version control P-280 (2018) 183–198.
- [86] S. Kelly, Collaborative modelling with version control, in: Software Technologies: Applications and Foundations - STAF 2017 Collocated Workshops, Marburg, Germany, July 17–21, 2017, Revised Selected Papers, Vol. 10748 of LNCS, Springer, 2017, pp. 20–29. doi:10.1007/978-3-319-74730-9_3.
- [87] J. L. C. Izquierdo, J. Cabot, Collaboro: a collaborative (meta) modeling tool, PeerJ Comput. Sci. 2 (2016) e84. doi:10.7717/peerj-cs.84.
- [88] M. A. Tröls, A. Mashkoo, A. Egyed, Multifaceted consistency checking of collaborative engineering artifacts, in: 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, Munich, Germany, IEEE, 2019, pp. 278–287. doi:10.1109/MODELS-C.2019.00044.
- [89] N. M. Preguiça, J. M. Marquês, M. Shapiro, M. Letia, A commutative replicated data type for cooperative editing, in: 29th IEEE International Conference on Distributed Computing Systems, Montreal, QC, Canada, IEEE, 2009, pp. 395–403. doi:10.1109/ICDCS.2009.20.
- [90] P. Nicolaescu, M. Rosenstengel, M. Derntl, R. Klamma, M. Jarke, Near real-time collaborative modeling for view-based web information systems engineering, Inf. Syst. 74 (Part) (2018) 23–39. doi:10.1016/j.is.2017.07.008.
- [91] R. Saini, G. Mussbacher, 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, 2021, pp. 35–44. doi:10.1109/MODELS-C53483.2021.00013.
- [92] S. N. Voogd, K. Aslam, L. van Gool, B. Theelen, I. Malavolta, Real-time collaborative modeling across language workbenches - a case on jetbrains MPS and eclipse spoofox (2021) 16–26doi:10.1109/MODELS-C53483.2021.00011.
- [93] G. Sunyé, Model consistency for distributed collaborative modeling, in: Modelling Foundations and Applications - 13th European Conference, Marburg, Germany, Vol. 10376 of Lecture Notes in Computer Science, Springer, 2017, pp. 197–212. doi:10.1007/978-3-319-61482-3_12.
- [94] F. Härer, 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, 2018, p. 55.
- [95] E. Syriani, H. Vangheluwe, R. Mannadiar, C. Hansen, S. V. Mierlo, H. Ergin, Atompm: A web-based modeling environment, in: Joint Proceedings of the 16th International Conference on Model Driven Engineering Languages and Systems, Miami, FL, USA, Vol. 1115 of CEUR Workshop Proceedings, CEUR-WS.org, 2013, pp. 21–25.
- [96] J. Gallardo, C. Bravo, M. A. Redondo, A model-driven development method for collaborative modeling tools, J. Netw. Comput. Appl. 35 (3) (2012) 1086–1105. doi:10.1016/j.jnca.2011.12.009.
- [97] L. A. Williams, Integrating pair programming into a software development process, in: 14th Conference on Software Engineering Education and Training, Charlotte, North Carolina, USA, IEEE, 2001, p. 27. doi:10.1109/CSEE.2001.913816.

- [98] J. E. Hannay, T. Dybå, E. Arisholm, D. I. K. Sjøberg, The effectiveness of pair programming: A meta-analysis, *Inf. Softw. Technol.* 51 (7) (2009) 1110–1122. doi:10.1016/j.infsof.2009.02.001.
- [99] A. Demuth, M. Riedl-Ehrenleitner, A. Nöhner, P. Hehenberger, K. Zeman, A. Egyed, Designspace: an infrastructure for multi-user/multi-tool engineering, in: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, Salamanca, Spain, ACM, 2015, pp. 1486–1491. doi:10.1145/2695664.2695697.
- [100] E. Carmel, R. Agarwal, Tactical approaches for alleviating distance in global software development, *IEEE Softw.* 18 (2) (2001) 22–29. doi:10.1109/52.914734.
- [101] L. Taxén, An integration centric approach for the coordination of distributed software development projects, *Inf. Softw. Technol.* 48 (9) (2006) 767–780. doi:10.1016/j.infsof.2006.01.007.
- [102] J. A. Espinosa, E. Carmel, The impact of time separation on coordination in global software teams: a conceptual foundation, *Softw. Process. Improv. Pract.* 8 (4) (2003) 249–266. doi:10.1002/spip.185.
- [103] D. Ford, T. Zimmermann, C. Bird, N. Nagappan, 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, 2017, pp. 394–403. doi:10.1109/ESEM.2017.54.
- [104] P. Mohapatra, P. Björndal, K. Smiley, Causal analysis of factors governing collaboration in global software development teams, in: *5th IEEE International Conference on Global Software Engineering*, Princeton, NJ, USA, IEEE, 2010, pp. 128–132. doi:10.1109/ICGSE.2010.23.
- [105] J. Y. Bang, D. Popescu, G. Edwards, N. Medvidovic, N. N. Kulkarni, G. M. Rama, S. Padmanabhan, 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, 2010, pp. 243–246. doi:10.1145/1810295.1810341.
- [106] S. Mirri, M. Rocchetti, P. Salomoni, Collaborative design of software applications: the role of users, *Hum. centric Comput. Inf. Sci.* 8 (2018) 6. doi:10.1186/s13673-018-0129-6.
- [107] J. B. Awotunde, F. E. Ayo, R. O. Ogundokun, O. E. Matiluko, E. A. Adeniyi, 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*, Vol. 12254 of LNCS, Springer, 2020, pp. 311–319. doi:10.1007/978-3-030-58817-5_24.
- [108] P. Mohagheghi, W. Gilani, A. Stefanescu, M. A. Fernández, An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases, *Empir. Softw. Eng.* 18 (1) (2013) 89–116. doi:10.1007/s10664-012-9196-x.
- [109] J. E. Hutchinson, M. Rouncefield, J. Whittle, Model-driven engineering practices in industry, in: *Proceedings of the 33rd International Conference on Software Engineering*, Waikiki, Honolulu, HI, USA, ACM, 2011, pp. 633–642. doi:10.1145/1985793.1985882.
- [110] M. Stephan, 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, 2019, pp. 240–247. doi:10.5220/0007255502420249.
- [111] K. Altmanninger, M. Seidl, M. Wimmer, A survey on model versioning approaches, *Int. J. Web Inf. Syst.* 5 (3) (2009) 271–304. doi:10.1108/17440080910983556.