



Human factors in model-driven engineering: future research goals and initiatives for MDE

Grischa Liebel¹ · Jil Klünder² · Regina Hebig³ · Christopher Lazik⁴ · Inês Nunes⁵ · Isabella Graßl⁶ · Jan-Philipp Steghöfer⁷ · Joeri Exelmans⁸ · Julian Oertel³ · Kai Marquardt⁹ · Katharina Juhnke¹⁰ · Kurt Schneider² · Lucas Gren¹¹ · Lucia Happe¹² · Marc Herrmann² · Marvin Wyrich¹³ · Matthias Tichy¹⁴ · Miguel Goulão¹⁵ · Rebekka Wohlrab¹¹ · Reyhaneh Kalantari¹⁶ · Robert Heinrich¹⁷ · Sandra Greiner¹⁸ · Satrio Adi Rukmono¹⁹ · Shalini Chakraborty¹ · Silvia Abrahão²⁰ · Vasco Amaral¹⁵

Received: 18 March 2024 / Revised: 18 April 2024 / Accepted: 20 April 2024 / Published online: 28 June 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Software modelling and model-driven engineering (MDE) is traditionally studied from a technical perspective. However, one of the core motivations behind the use of software models is inherently human-centred. Models aim to enable practitioners to communicate about software designs, make software understandable, or make software easier to write through domain-specific modelling languages. Several recent studies challenge the idea that these aims can always be reached and indicate that human factors play a role in the success of MDE. However, there is an under-representation of research focusing on human factors in modelling. During a GI-Dagstuhl seminar, topics related to human factors in modelling were discussed by 26 expert participants from research and industry. In breakout groups, five topics were covered in depth, namely modelling human aspects, factors of modeller experience, diversity and inclusion in MDE, collaboration and MDE, and teaching human-aware MDE. We summarise our insights gained during the discussions on the five topics. We formulate research goals, questions, and propositions that support directing future initiatives towards an MDE community that is aware of and supportive of human factors and values.

Keywords MDE · Modelling · Modeling · Human factors · Workshop

1 Introduction

The use of software models to design, analyse, generate, and document software, often called model-driven engineering (MDE), is at the core of software engineering (SE) research and practice. Strong communities have been built around modelling technologies, as evidenced by several international conferences specialising in the topic, e.g. the CORE A-ranked Models conference.

Models are aimed at enabling practitioners to communicate about software designs [1], at making software understandable [2], or at making software easier to write through domain-specific modelling languages [3]. The modelling community traditionally studies these aspects from a technical perspective, considering aspects such as the technical design of language workbenches, specifications of

modelling standards, grammar generation from meta models, and model transformation languages. However, one of the core motivations behind the use of software models is inherently human-centred, and several recent studies indicate that human factors play a role in the success of MDE [4, 5]. Such research focusing on human factors in modelling is currently under-represented, including topics such as:

- Human factors that influence the way models are used;
- Design of modelling languages and technologies that account for human factors, such as cognitive ergonomics and accessibility issues;
- The potential of models in supporting humans in SE, e.g. during software comprehension, for explainable AI, or usability design.

To address this gap in existing work, we gathered a group of 26 researchers related to human factors and modelling research in Schloss Dagstuhl in November 2023 to construct

Communicated by Bernhard Rumpe.

Extended author information available on the last page of the article

tively discuss ways forward in research related to human factors in MDE. In the following, we summarise the discussions during the seminar, as well as the recommendations and suggested actions resulting from them.

2 Overview of discussed topics

The insights presented in this paper emerge from discussions at a GI-Dagstuhl seminar on "Human Factors in Model-Driven Engineering".¹ Details about the participants of the seminar can be found in the Appendix A.

Using the 1-2-4-ALL microstructure², we elaborated goals, outcomes, and topics to be discussed over the week. In this practice, participants are encouraged to individually brainstorm specific topics for a limited amount of time, such as the goals, outcomes, and topics for the seminar. This step is followed by pairing up with another participant, comparing the individual results and agreeing on joint results. The same is repeated with groups of four and, finally, with all participants.

In the first step, the participants elicited their goals and expected outcomes of the workshop with the 1-2-4-ALL microstructure. The most frequently mentioned (and aligned) goals were (1) to formulate a research agenda, respectively, to build a roadmap, (2) to exchange ideas, (3) to make MDE more practical, (4) to unify projects, (5) to develop ideas for empirical research on MDE in industry, and (6) to use more models (of humans).

The requested outcomes of the workshop were quite manifold, but all participants agreed upon establishing contacts and collaborations for specific subtopics and finding new friends. From a research perspective, the mentioned outcomes included writing joint papers, research proposals, or a book; finding approaches to integrate human factors in software engineering; to establish concepts to integrate MDE in development workflows; to identify best practices for research in industrial settings; to find strategy guidelines for teaching MDE; and to develop a new paradigm for human behaviour.

In the next step, the group of participants identified a total of ten topics to be discussed over the week:

1. Factors of Modeller Experience
2. Collaboration and MDE
3. Diversity and Inclusion in MDE
4. Modelling Human Factors
5. Teaching Human-aware MDE
6. Research Methods
7. AI complementing MDE

8. MDE ecosystem maturity
9. Usability in live modelling
10. Sustainability and well-being

As ten topics were considered too many to be discussed in detail over a week, and as the number of goals and outcomes was too large, a voting with three votes per participant helped prioritise the topics. This resulted in the final five topics (Topics 1 to 5 in the list above) being discussed throughout the week. The same voting also allowed us to condense the outcomes into:

- New active contacts
- Concepts to embed MDE in development workflows
- Best practices for research in industrial settings
- A collaborative paper

The identified two main goals are (1) to elicit ideas for empirical research on MDE in industry and (2) to use more models of and for humans.

In the following sections, we will describe the outcomes of the discussions surrounding each topic in more detail.

3 Topic 1: factors of modeller experience

In the breakout group for Topic 1, we discussed the factors that contribute to the Modeller Experience (MX). The motivation was to recognise MX's key role in MDE adoption and its lack of detailed definition in scholarly work. Our discussion aimed to systematise MX by identifying and defining its contributing factors within MDE.

A fundamental insight is that MX varies in organisations' modelling contexts. For example, those using formal MBSE approaches, such as in the automotive sector, differ markedly in their modelling practices compared to organisations using informal modelling processes. This has led us to recognise four key modelling workflows, each reflecting distinct aspects of MX in industrial environments: 1) Low-code; 2) Everything-as-code; 3) MBSE in the Automotive Domain; and 4) Informal Modelling. We have only considered the last three of these workflows in this work.

The main objective of this work is to address the following questions: What factors contribute to MX in the different modelling workflows identified? By answering this question, we hope to better understand the needs of practitioners based on their type of workflow and, consequently, be able to provide better solutions in our research. We also believe that characterising modelling solutions using our workflows and factors will make it easier to define a target audience and delineate the research scope.

To answer this question, a focus group of ten modelling experts from academia and industry brainstormed using

¹ <https://www.dagstuhl.de/23473>.

² <https://www.liberatingstructures.com/1-1-2-4-all/>.

```
graph LR; OM[Organisational-external Motivation] --> LC[Language Choice & Number of Languages]; OM --> C[Collaboration & Versioning]; OM --> T[Technical Features]; OIM[Organisational-internal Motivation] --> LC; OIM --> C; OIM --> T; IM[Intrinsic Motivation] --> C; IM --> T; T[Training] --> C; T --> TFX[Technical Features]; T --> LC; T --> LCP[Language Complexity]; C --> I[Integration]; C --> TFX; TFX --> I; TFX --> LCP; I --> LC; I --> LCP; LC --> LCP;
```

The flowchart illustrates the conceptual model of factors influencing language choice and complexity. It shows how various motivational and training factors lead to collaboration, tool UX, and technical features, which in turn influence integration and language complexity.

- Organisational-external Motivation** (blue box) influences **Language Choice & Number of Languages** (orange box), **Collaboration & Versioning** (green box), and **Technical Features** (green box).
- Organisational-internal Motivation** (blue box) influences **Language Choice & Number of Languages** (orange box), **Collaboration & Versioning** (green box), and **Technical Features** (green box).
- Intrinsic Motivation** (blue box) influences **Collaboration & Versioning** (green box) and **Technical Features** (green box).
- Training** (blue box) influences **Collaboration & Versioning** (green box), **Technical Features** (green box), **Language Choice & Number of Languages** (orange box), and **Language Complexity** (orange box).
- Collaboration & Versioning** (green box) influences **Integration** (green box) and **Technical Features** (green box).
- Technical Features** (green box) influences **Integration** (green box) and **Language Complexity** (orange box).
- Integration** (green box) influences **Language Choice & Number of Languages** (orange box) and **Language Complexity** (orange box).
- Language Choice & Number of Languages** (orange box) influences **Language Complexity** (orange box).

 Springer

tions of the same models attempts to address these issues [22] fundamentally.

Technical capabilities The capabilities resulting from the combination of the modelling language and tool to, for example, reason about the modelled system's properties or generate downstream artefacts.

3.3 Non-technical factors

Modeller-intrinsic motivation is driven by perceived benefits and positive emotions from modelling and leads modellers to use modelling. *Intrinsic motivation* stands in contrast to extrinsic motivation, where the incentive originates from an external source rather than the inherent appeal of the task [23]. Intrinsic motivation is a key predictor of developer experience in software development [24].

Organisation-intrinsic motivation Internal motivations within organisations significantly influence the adoption of modelling, driven by perceived benefits in productivity, quality, cost, and collective well-being. The literature highlights these benefits, particularly in the embedded software industry where efficiency and quality are key adoption motivators for MDE [25]. Additionally, factors like organisational culture, expertise, and internal advocacy significantly influence MDE adoption [26].

Organisation-extrinsic motivation External factors influencing an organisation's adoption and approach to modelling include standards, regulations, tool availability, and customer demands. Adherence to regulations, especially in the embedded systems industry, is a significant driver for adopting MDE [27].

Training Factors related to skills and knowledge in using modelling languages and tools. The importance of training for MDE adoption is highlighted [26], with challenges in the *insufficiency of training resources* [28] and *developer training efforts*.

The working group believes that these factors are important aspects that affect MX and that understanding their relationships is crucial to understanding why modelling works (or does not) in an organisation or project and how to improve MX. To test this idea, we will apply these factors to the workflows already mentioned above. In these workflows, we captured typical well-known scenarios from practice and literature: informal modelling with and without digitally captured diagrams, architectural modelling in the classical model-based systems engineering sense in the automotive industry, and modelling as part of work on infrastructure-as-code. We believe that these scenarios are archetypes and that, by applying the factors to them, we can learn more about their applicability and gain insights into their relevance.

As a result of this effort, we envision some possible future research questions:

- Is there evidence that the proposed MX factors are relevant in practice? While we brought together significant experience in modelling from practice and research, our current understanding of the factors and their relationships should be validated in industrial settings with the help of practitioners. To achieve this, we plan to conduct an empirical study, possibly using a mixed-method design with interviews and a survey.
- Do existing modelling tools address the factors we have identified? On the one hand, answering this question will allow us to validate the relevance of the proposed factors. On the other hand, this mapping will also allow us to identify factors unsupported by a modelling approach or tool and are, therefore, candidates for further exploration.

A forthcoming paper that describes the factors in more detail and maps them to the archetypical workflows will be a first step towards answering these questions.

4 Topic 2: collaboration and MDE

In the breakout group for Topic 2, we discussed the topic of collaboration and MDE, which may be parsed in at least three ways: First, we can associate *modelling together* to solve a problem, such as with interactive editors or version control systems trimmed for supporting modelling in teams. Second, we can associate using models to *express* collaborative development and, third, using models to *leverage communication and, thus, collaboration* (between different roles in a software project). While the first two options may provide several possibilities for research, in the sequel, we focus on the third direction of using models to support communication in software projects.

4.1 Motivation

In software projects, different roles and teams need to exchange information. Although models provide an abstraction layer based on which humans in different roles can communicate, models are not sufficiently used to *share* information between roles. Since they provide abstractions which are trimmed to the purpose of the role, another role may not understand them. Thus, the missing capabilities of models for collaboration in software projects provoke lost, inconsistent, or untraceable information, such as requirements, design, and implementation decisions. The following example illustrates these issues.

4.1.1 Motivating example

Let us consider the transition from the requirements engineering to the architecture and design phase: requirements

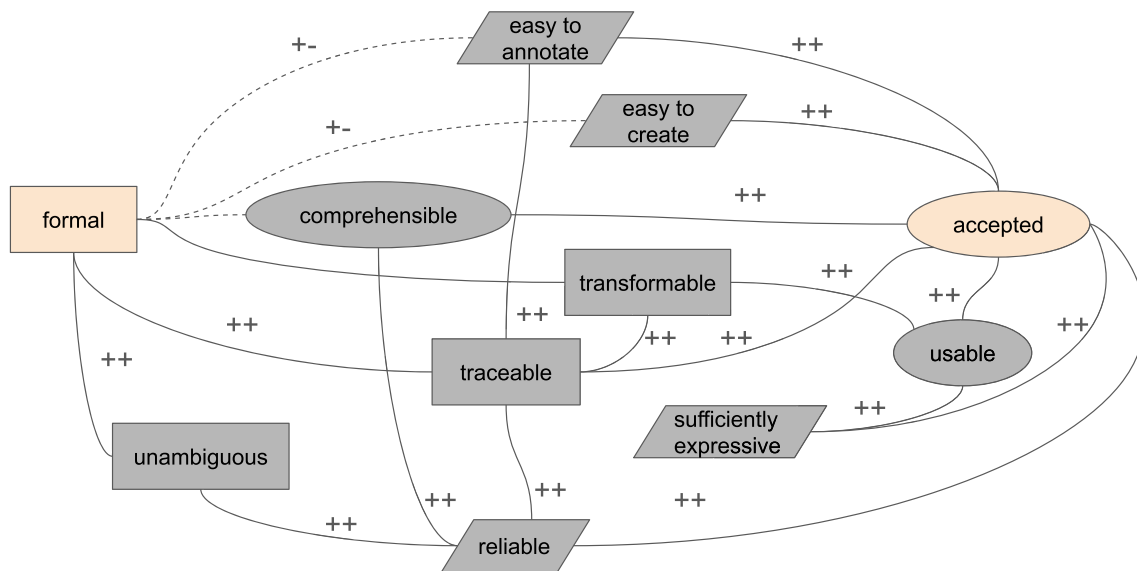


Fig. 2 Characteristics of models to benefit communication in collaborations

analysts should provide information on the results of their work to the software architect. Understanding which requirements must be satisfied and what they mean is crucial for project success. Consequently, it is essential that requirements analysts and software architects (and later on any other involved person in the process, such as developers, testers, etc.) form the same understanding of the requirements' meaning.

In case of a plan-driven process, a specification with use cases and use case diagrams results from the requirements engineering phase, representing one possibility to model the elicited requirements. Although use case diagrams are an established means of modelling requirements, it is debatable whether they suffice for software architects to understand for what software they should provide an architecture. Instead, both roles communicate informally and verbally, which always introduces the risk of misunderstandings or lost information.

Using persisted models provided by the requirements engineers, such as use case diagrams, can support establishing a shared understanding, but in practice, they tend to be neglected, particularly in communicating between project roles [29, 30].

4.2 Characteristics of models for supporting collaboration

To solve the problem of refused communication between roles using models, we first need to examine which characteristics of models hinder or benefit communication and, thus, collaboration between different roles. In the following, we present key characteristics that we elicited in extensive

focus group discussions and refined based on discussions with all participants.

First, two main characteristics of models are decisive, the formality and the acceptance state, as depicted in orange in Fig. 2. We consider “acceptance” as the goal (necessary requirement) to be satisfied when we want a model to be used for communication. We identified the “formality” of the model as the independent variable which we can control. Although formality does not directly influence acceptance, several mediating factors exist that formality influences (e.g. the ease of use) and which, in turn, influence acceptance.

Furthermore, the figure distinguishes the characteristics of a model that are technical-centred (visualised as rectangles) and human-centred (visualised as ellipses), and some that are shared by both (visualised as parallelograms). The association of an attribute to the human or the technical support for the model depends on the measurement. Human-centred characteristics completely depend on the human's assessment (and, hence, vary depending on the person interacting with the model), whereas one should be able to measure technical-centred attributes objectively. The figure presents relationships between the different characteristics with clear increasing or decreasing effect (“the more X, the more Y”, “the less X, the less Y”; denoted as “++”) or with contradicting effect (“the more X, the less Y” or “the less X, the more Y”; denoted as “+-” and vice versa, respectively). For example, the more formal a model is, the more complex it is to create, as it requires more effort to define it properly. Conversely, for instance, the more comprehensible a model is, the contained information is of higher reliability which, in turn, positively affects the acceptance of a model.

In summary, based on the figure we can conclude that the amount of formalism influences negatively the ease of anno-

tation, creation, and comprehension. In contrast, the more formal a model is, the higher the likelihood it is unambiguous, transformable, traceable and, in turn, reliable and usable, positively affecting its acceptance.

4.3 Research agenda

As a result of eliciting these characteristics, we propose to pursue the following steps to optimise using models for effective communication and collaborations:

1. Investigate why different roles do not use models for communication across their field.
2. Empirically elicit characteristics in frequently used models which complicate or ease the communication across roles and stakeholders.
3. Compare the characteristics with human needs to communicate provide guidelines that enhance existing models to facilitate communication across roles
4. Derive metrics for measuring the quality of the enhanced models.

By pursuing these steps, we will first derive insights from practice and research, which may require interdisciplinary models. Furthermore, the elicited characteristics allow us to define guidelines to optimally use models in communication. Eventually, providing metrics may increase the practical usage of models in communication across heterogeneous roles in software projects.

5 Topic 3: diversity and inclusion in MDE

In the breakout group for Topic 3, we critically discussed diversity and inclusion as related to MDE and formulated a research roadmap for the topic. The diversity dimensions we consider are age, educational background, cultural background, gender, experience level, cognitive diversity (i.e. neurodiversity), and colour blindness of the modeller and the model end user. We either have evidence suggesting differences concerning MDE for these dimensions, or we hypothesise a potential impact. For instance, cognitive diversity, user age, gender, or cultural background might all affect preferred working methods and problem-solving styles. In turn, these preferences might align well or conflict with MDE tools, notations, and workflows.

Several discussions and brainstorming sessions based on these diversity dimensions led us to six distinct research directions related to Diversity and Inclusion in MDE. An overview of these directions and their connections is depicted in Fig. 3. In the following, we discuss each direction separately.

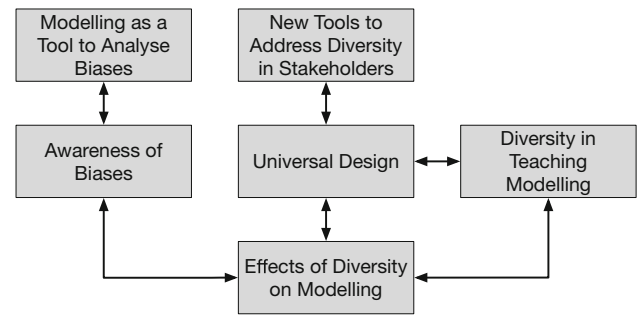


Fig. 3 Overview of identified dimensions of research topics relevant for diversity in MDE

5.1 Universal design

Universal Design is “the design and composition of an environment so that it can be accessed, understood and used to the greatest extent possible by all people regardless of their age, size, ability, or disability”.⁴ We transfer this concept to MDE, proposing that a better understanding of what needs different diverse groups have will lead to improvements in MDE and related practice. Improving modelling tools, languages, techniques, and practices (MTLTPs) accordingly will ultimately benefit all people and the field, not only the specific group targeted.

Thus, the central goal of this research direction is to understand the needs for MTLTPs with respect to several diversity dimensions. To address this goal, we need to investigate what aspects of modelling are affected by diversity concerns, and which barriers exist for diverse stakeholders. This relates to tooling and visual aspects, e.g. modelling notations and layout, but also comprises cognitive aspects, such as preferred workflow, model comprehension, and information processing style. Understanding these effects and the involved dimensions will allow for the creation of guidelines for better tool and practice design surrounding MDE that benefit all users.

5.2 Awareness of biases

MTLTPs are purposefully designed but may contain biases, which raises concerns about their inherent subjectivity and their reflection on the modeller’s background.

We hypothesise the presence of biases in MTLTPs towards specific diversity dimensions, highlighting the importance of studying these biases. The objective is to compile scenarios illustrating how MTLTPs might yield biased results, providing insights for refining design guidelines.

For instance, cultural biases may emerge in language design, affecting text length and layout preferences. In mod-

⁴ <https://universaldesign.ie/what-is-universal-design/>.

elling videos from India, we observe that reference names and annotations tend to be longer than the common European approach, which might relate to low & high context cultures [31]. Additionally, gender bias is noticeable in teaching materials through stereotypical representations, potentially contributing to inequities [32].

The overarching objective of this dimension is twofold: (1) make biases explicit in models and (2) raise awareness to guide future MTLTP design. This involves examining real-world examples of biases from diverse literature, assessing student assignments and critically evaluating teaching materials, papers on modelling, and content analysis of teaching videos.

5.3 Effects of diversity on modelling

The exploration of the effects of diversity on modelling centres on recognising that models are created, maintained, and interpreted by individuals with unique characteristics, influencing their interactions with models.

Given the impact of diverse group compositions on communication and collaboration methods, the primary objective is to understand better how various diversity dimensions influence the modelling process within group dynamics, and subsequently affect the features of resulting models.

Examples from other SE sub-fields, such as programming, demonstrate that individuals from different age groups [33] or genders [34] exhibit varied working approaches, stress coping mechanisms, and conflict resolution strategies, influencing the outcomes, such as code quality [34].

The long-term goal involves creating awareness of expected differences, developing inclusive modelling tools, and enhancing accessibility in modelling explanations through comparative analysis of modelling discussions and decisions among diverse groups via participatory observations.

5.4 Modelling as a tool to analyse biases

Models are a tool for abstraction. As such, we hypothesise that they may be more suitable for analysing and detecting certain biases that may be otherwise obfuscated in more detailed artefacts defined at lower abstraction levels, such as source code. This research direction aims to understand how we can leverage models to detect potential biases. This could be done by creating methods for systematically reasoning about diversity through models, or by using models in the context of participatory design to involve diverse stakeholders in the modelling process.

For example, reasoning about requirements models may facilitate detecting missing requirements that affect marginalised or underrepresented groups. In addition, models can also be created to help detect bias or avoid it [35].

Finally, models may be used to analyse potential biases in datasets built for machine learning, which could affect data collection, assumptions about the data, specific valid scenarios, or data quality.

5.5 Diversity in teaching modelling

The effective teaching of MDE and modelling requires consideration of various diversity dimensions and their implications for modelling practices. Recognising and mitigating barriers to learning and adopting modelling techniques is crucial to fostering an inclusive and accessible educational environment.

To overcome these barriers, tailored teaching strategies and methodologies are crucial. For example, enhancing self-efficacy in female students has been shown to be essential for engaging them in computer science disciplines, including MDE [36]. Furthermore, the absence of timely feedback during the learning process can result in frustration, adversely affecting students' willingness to embrace and learn modelling. Students on the autistic spectrum may encounter difficulties with free-hand modelling exercises, owing to a need for clear processes and well-defined rules [37].

Addressing the needs of diverse groups during teaching can potentially reduce barriers to adoption. Key questions for exploration include how teaching methods and tools impact different groups, identifying teaching strategies that may conflict with diverse needs, and understanding how to mitigate conflicts arising from diversity within student groups to ensure a positive learning experience. The long-term goal is to develop and promote inclusive MDE teaching practices through collaborative efforts involving representatives of diverse groups. Evaluating developed teaching methods on a larger scale, including international perspectives, will be essential to gauge effectiveness and ensure inclusion.

5.6 New tools to address diversity in stakeholders

Enabling more diversity in modelling also asks for a technical view on MDE. Tools and languages, including Domain-Specific Languages (DSLs), provide the basis for modelling. Thus, the tools and modelling languages in use need to be designed to be inclusive of the needs of diverse users. If tools are not designed according to the requirements of a diverse group of users, they introduce additional obstacles to the modelling process [35].

We hypothesise that the literature already provides substantial approaches that offer potential solutions. For example, the dedicated creation of DSLs or Blended Modelling [38] environments for specific stakeholders or domain experts can be further used to address stakeholder diversity as well. By offering a variety of concrete syntaxes related to the same abstract syntax, different groups can choose the

concrete syntax that successfully addresses their particular needs.

To succeed in the goal of providing tools and modelling languages that are inclusive of the needs of diverse users, we propose studying the following high-level questions:

- How can modelling tools successfully address the needs of diverse groups?
- How can tools address collaborative modelling when users prefer different syntaxes/representations?
- How can blended modelling languages or DSLs be designed for the use of diverse groups?

The long-term goal of the research on this issue should be to develop guidelines limiting the barriers to using modelling tools. These guidelines can also help evaluate newly developed tools on their inclusiveness for diverse groups.

6 Topic 4: modelling human factors

In the breakout group for Topic 4, we discussed that when modelling human factors, many possible aspects could be considered. We strived for an interdisciplinary topic between software engineering and psychology to find a concrete aspect that we could work on during the seminar.

6.1 Motivation

Over the last few years, the awareness of human issues and individual human aspects in SE has grown. There are currently specialised development processes (such as human-centred design) that emphasise human involvement. Therefore, we discussed what other aspects of humans could be integrated into our existing approaches. Previous psychology research has aimed to model or represent different aspects influencing human behaviour. Humans are complex beings, which makes it difficult to create holistic models. However, individual human aspects can be modelled currently with sufficient significance.

6.2 Human values

One of these aspects are human values, as proposed within Schwartz's theory of basic human values [39]. These values are modelled in relation to each other, as depicted in Fig. 4, where nearby values are related, whereas bipolar values are conflicting.

From the SE perspective, human values are crucial. Human values of end users play an important role in software projects: values such as trustworthiness, fairness, etc., are becoming increasingly important, and companies therefore want to be associated with such positive values. Until



Fig. 4 Circular model of Schwartz's theory of basic human values. (Source: Schwartz et al. [39])

now, human values have hardly been explicitly taken into account in the development process (e.g. as non-functional requirements).

6.3 Process

We envision a human value-centric software process that deliberately consists largely of familiar building blocks but puts them together in a new way. This is because the human values of end users are often vaguely formulated at the beginning and have different meanings for different people. Therefore, in the development process, these values must be captured during the requirements elicitation and passed on to the stakeholders and developers. Traceability is crucial so that human values do not remain an abstract concept but are linked to design and implementation artefacts.

6.4 Novelty

Building on existing concepts such as user involvement and personas, our envisioned approach aims to transcend individual human desires to focus on emerging human values, using the Schwartz model as an example. It provides a pragmatic technique for constructing personas as a collection of human aspects, focusing on human values. The approach suggests a systematic progression from high-level needs to decisions that guide the implementation of value-related requirements. It proposes a systematic framework suitable for ordinary software organisations. The approach is not limited to human-centred design professionals but also con-

siders developers, business owners, and other stakeholders to address the challenges related to the ambiguity of human values. Additionally, we are developing important research questions to enhance and evaluate this approach effectively.

6.5 Directions

After the seminar, we worked on a systematic approach to represent human values and ensure traceability. In the near future, we will publish a separate research article that describes the detailed approach.

7 Topic 5: teaching human-aware MDE

Models or sketches for thinking are universally beneficial due to their ability to clarify complex ideas and improve understanding. In the realm of software and systems engineering, models serve as essential tools for simplifying complex systems, enabling a shared language among stakeholders, enhancing collaboration efficiency, and improving decision-making processes. Effective modelling, especially in open-ended problems, necessitates structured thinking and a clear purpose, ensuring that all aspects of the system are considered and appropriately addressed.

Teaching human-aware MDE goes beyond the dissemination of tools and techniques; it involves nurturing a mindset geared towards prioritisation, in-depth analysis, and strategic approaches to complex issues. This section discusses the educational frameworks and pedagogical strategies essential for cultivating such a mindset.

7.1 Pedagogical challenges

7.1.1 The confidence in modelling skills

A key challenge in teaching modelling skills is students' lack of confidence in their modelling abilities. This often stems from a fear of making mistakes or not having complete knowledge of the modelled domain, leading to scepticism towards their models. This scepticism towards models can be significantly mitigated by teaching structured thinking, which provides a clear, strategic framework for tackling modelling problems.

A significant problem faced by computer science students is the premature focus on implementation during initial designs, which inhibits the development and utilisation of abstraction skills. This focus on implementation is a manifestation of difficulties in separating problem space and solution space [40].

It is crucial for educational approaches to focus on exploring the problem space and abstract thinking. Abstraction in modelling involves the process of identifying and focusing

on the essential aspects of a real-world problem or system while disregarding the less significant details. This selective focus allows modellers to create simplified representations (models) of complex systems that retain only the features pertinent to the problem at hand. The goal is to make the problem more manageable and the system more understandable without oversimplifying it to the point of losing essential characteristics.

In contrast to the typical computer science view where complexity is hidden in nested levels of abstraction, abstraction in modelling is driven by the need to explore and understand the problem space. This involves:

- **Problem space exploration:** Understanding the various elements, relationships, and processes that constitute the system. This step is crucial for identifying what should be included in the model and what can be left out.
- **Purpose-relevant information extraction:** Determining which aspects of the system are relevant to the goals of the modelling effort. This means different models might be created from the same system for different purposes, each abstracting different facets or details based on what is most relevant to the problem or question being addressed.
- **Creation of usable models:** Developing a representation of the system that is both simpler than the reality but still retains all necessary detail to be effective for solving the problem or answering the question at hand. This involves a balance between simplification and accuracy, ensuring that the model is both tractable and valid within the context of its intended use.

Abstraction, therefore, is a critical skill in modelling because it allows the modeller to construct meaningful and usable models without becoming overwhelmed by the complexities of the real world. It involves a deliberate process of filtering out noise and focusing on the essence of the problem, which requires a deep understanding of both the system being modelled and the context in which the model will be applied. This skill enables modellers to navigate the vast problem spaces they encounter and develop effective solutions and strategies. This will better prepare students to contribute effectively in the rapidly evolving technological landscape. Abstraction skills are difficult to teach as they require balancing detail and generalisation.

7.1.2 The crucial role of feedback

Feedback is an indispensable component in the learning process, as it supports learning and builds confidence. The lack of relevant and constructive feedback contributes to the challenges students and teachers face. Furthermore, the previously mentioned lack of confidence of students in their modelling abilities is often exacerbated by the absence of

relevant and constructive feedback on the quality of their models.

However, providing quality feedback on models presents unique challenges. For example:

Assessing completeness: In complex domains, it can be difficult to determine if a model comprehensively covers all relevant concepts.

Evaluating correctness: The correctness of a model goes beyond just accurately representing individual concepts; it also involves the relationships between these concepts.

Navigating multiple valid models: For a given problem within a domain, there can be several different yet equally valid models. This complicates the feedback process as there is no single “right” model against which students’ work can be benchmarked.

Understanding model quality: There is no tangible, universally accepted definition of model quality; hence providing feedback on model quality can be abstract and challenging for students to grasp.

Addressing these challenges necessitates a nuanced approach to feedback that recognises the complexities involved in modelling.

7.1.3 The embracing of and motivation for modelling

Many students find it difficult to appreciate the significance of modelling. Modelling can be seen as an overwhelming abstract and annoying task, being in the way to start programming. This view can lead to a lack of interest in mastering modelling techniques and recognising their application across various fields.

The true benefit of modelling manifests itself in collaborative, iterative, and long-term projects. However, these situations are rarely utilised in education. For instance, while group projects are common, they rarely focus explicitly on fostering collaborative modelling skills. Additionally, the typical duration of university courses, spanning from a few weeks to a semester, limits the depth of engagement students can have with complex modelling projects, thus restricting the potential for meaningful teamwork and iterative learning.

Finally, our observations suggest that the ability of students to read and comprehend existing models is underdeveloped. This deficiency stems from educational practices that focus on creating new models, neglecting the equally important skill of model interpretation. Students seldom receive assignments that require them to analyse and understand existing models, which leads to a lack of familiarity and confidence in dealing with pre-established frameworks. Furthermore, this leads students to adopt a skewed understanding of modelling. This gap undermines their confidence in mod-

elling and impairs their ability to effectively contribute to ongoing projects or adapt existing models to new contexts.

7.2 Addressing the challenges

Addressing the challenges outlined above requires an integrated approach to teaching modelling that highlights practical benefits, encourages active collaboration, incorporates activities and assignments that focus on the critical analysis of existing models, and demystifies the various techniques and notations used.

7.2.1 Strategies for assessing modelling skills

The challenge in teaching modelling skills is finding an equilibrium between the need for tangible, assessable outcomes and maintaining the holistic nature of modelling, which is critical for students’ comprehension and application in real-world scenarios. Educators must navigate this balance to cultivate both technical competence and an appreciation for the value of modelling in the broader interdisciplinary educational context.

To address this challenge, educators should consider implementing a multifaceted approach that includes incremental learning and constructive feedback on modelling assignments, structured thinking exercises that promote abstract reasoning and problem decomposition, and practical training through real-life case studies and examples. This approach, including critical essays, analysis papers, Socratic discussion seminars, peer reviews, and case studies, fosters a more profound understanding of the importance and applicability of modelling in various contexts.

7.2.2 Strategies for delivering constructive and relevant feedback

It can be beneficial to follow a structured guideline when assessing student models to elevate the quality of students’ models and provide better feedback. A key element of this strategy involves differentiating feedback between syntax and semantics. By distinguishing between these two dimensions, educators can provide more targeted feedback that addresses student models’ form and substance. Moreover, feedback should be positive and constructive, highlighting what students have done well and pointing out areas for improvement. This approach encourages learning by reinforcing successful strategies and providing clear, actionable suggestions for addressing weaknesses. Such feedback fosters a learning environment that values growth and clear, targeted improvement.

Feedback must be specific and refer to concrete instances within the model to be effective. For example, rather than making a general statement about the model’s organisation,

an instructor might point out how the placement of a particular element enhances the model's clarity and coherence. Additionally, feedback should clearly differentiate between the problem space and the solution space. This distinction helps students understand whether issues arise from their understanding of the problem or the resolution methods they have chosen to solve.

When measuring the quality of student models, visual aesthetics and the layout of diagrams can serve as an initial proxy for model quality, as they often reflect the student's attention to detail and organisational skills. However, one should be careful not to overemphasise form over function. The naming of model elements is also critical, as names should be descriptive, concise, and appropriate to the level of abstraction while also conforming to established conventions.

Finally, an attempt at completeness should be made by comparing the list of concepts in the problem description with those present in the model, identifying any missing or unnecessary elements. Educators can significantly enhance students' modelling skills through such thorough and thoughtful feedback, providing them with the tools and understanding necessary to develop high-quality, meaningful models.

7.2.3 Strategies for teaching structured thinking

Structured thinking is a foundational skill for modellers, fostering confidence in their models and underlining the relevance of modelling. It enables the breakdown of complex problems into manageable parts, facilitating clear analysis. Teaching structured thinking involves introducing students to frameworks and methodologies that guide the systematic exploration of problems and the development of coherent, logical models. Techniques such as mind mapping, decision trees, and logical frameworks can be instrumental in this process.

This educational approach should focus on iterative learning, where students iteratively refine their critical and strategic thinking abilities. Encouraging reflective practice, where students evaluate their thought processes and modelling decisions, can significantly enhance their understanding and application of structured thinking in modelling. Exercises designed to simulate real-world scenarios can provide practical experience in applying structured thinking, thereby enhancing students' confidence in their modelling abilities and the perceived value of modelling.

8 Discussion

The topics summarised in Sects. 3 to 7 outline various topics, goals, and research questions for future work. The five topics cover specific aspects concerning those topics in depth.

However, there are various connection points between the five topics that are worth highlighting. These connections are sketched in Fig. 5.

Overall, we observe that topics 1, 2, 3, and 5 are well-connected. Topic 1, *Factors of Modeller Experience*, provides a broad frame of technical and non-technical factors that can affect modeller experience. Topic 2, *Collaboration and MDE*, and Topic 3, *Diversity and Inclusion in MDE*, consider specific cases of such factors, i.e. diversity-related factors and communication/collaboration-related factors. These specific cases might give rise to additional MX-related factors, or moderate their effect. Jointly, the three topics promote the message that considering technical factors alone is insufficient for successful use and adoption of MDE, and to understand how and why MDE is applied in practice. These observations lead us to the following proposition:

Proposition 1 *We propose that modelling is not an objective process, but various technical and non-technical factors affect the individual modelling experience. The community needs to (1) investigate what these factors are and (2) investigate to what extent the factors affect modelling practice and adoption.*

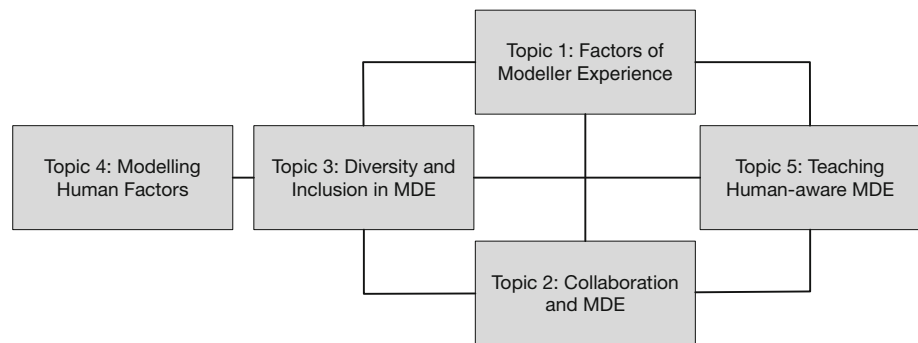
Topics 1 to 3 also all discuss that specific contexts need to be distinguished and studied separately, which is introduced as archetype workflows in *Factors of Modeller Experience*. While named differently, specific examples of such workflows are covered in *Diversity and Inclusion in MDE* with respect to diversity & inclusion, and in *Collaboration and MDE* with respect to collaborative aspects of MDE. Collecting such archetype workflows could be an important goal for the MDE community as a whole, as it would allow to better reason about the (external) validity of various empirical studies. We summarise this as follows:

Proposition 2 *We propose that various different contexts or workflows need to be considered in MDE research to allow for more precise conclusions to be drawn. The community needs to collect and validate representative workflows of the application of MDE in practice.*

Jointly addressing Propositions 1 and 2 could help the MDE community increase the validity of research results, for example, in reasoning about success factors of MDE-related interventions in industry.

Topic 5, *Teaching Human-aware MDE*, connects naturally to Topics 1 to 3 as, ultimately, students (and future engineers) need to be taught the complexity of modelling beyond purely syntactical and technical factors. On the one hand, this necessitates that educators are aware of relevant human factors and dimensions that affect MX so that MX does not negatively affect the classroom, ultimately leading to a lower acceptance

Fig. 5 Overview of the seminar topics



of MDE. On the other hand, it also requires highlighting these factors to the student population to ensure transfer to and awareness in practice. We summarise these observations as follows.

Proposition 3 *Education in MDE needs to mirror research efforts in the sense that educators (1) need to understand the effect of human factors on modeller (and student) experience, and (2) need to convey the importance of these human factors to the student population, so that knowledge transfer to practice takes place.*

Topic 4, *Modelling Human Factors*, is somewhat disconnected in contrast to the close coupling of the remaining topics. This group takes a different approach, namely how models can be used to analyse or consider human factors in SE. Topic 3, *Diversity and Inclusion in MDE*, covers this angle as well in two sub-points, i.e. that models can be a means to better understand diversity-related biases, and that existing contributions in the MDE community can be used to better support diverse needs among modellers and consumers of models in practice. This angle can be compared to previous discussions on how knowledge in the MDE community can serve other research fields, e.g. for climate modelling [41, 42]. However, in these previous cases, authors have advocated applying MDE-related knowledge to better understand phenomena in the scientific disciplines of engineering and natural sciences. Instead, we raise the point that the many contributions of the MDE community and their technical sophistication can and should be used for human factors. Overall, we propose the following.

Proposition 4 *Human factors are increasingly important in SE research and practice. Therefore, the community needs to apply their MDE-related knowledge and experience to models of human factors in SE.*

In addition to the research goals and questions suggested by the individual groups, and the propositions made above, connections between individual topics can lead to further interesting avenues for future work. For instance, Topic 2,

Collaboration and MDE naturally connects to several of the other topics for issues concerning collaboration or teamwork. Specifically, *Modeller Experience*, as covered in Topic 1, might vary from individual model use in cases of collaborative modelling. *Versioning* is highlighted as a technical factor contributing to MX and essential for successful collaborative MDE. However, further technical and non-technical factors could similarly affect MX in collaborative settings. Finally, *Collaboration and MDE* has a joint connection to Topic 3, *Diversity and Inclusion in MDE*. Collaborative modelling in a diverse group of people can lead to diverse, and potentially conflicting viewpoints, opinions, and preferences. This influence of diversity needs to be better understood to enable successful collaborative modelling practices. Similarly, in education, collaborative modelling in diverse groups needs to be learned and practised to enable university graduates to deal with this diversity in practice. Topic 1, *Factors of Modeller Experience*, has an additional connection to *Diversity and Inclusion in MDE*. That is, several non-technical factors related to MX might be affected by the diversity of the modeller(s) or the model's users. For example, preferences on factors such as layout, tooling, or amount of information contained in a model can potentially affect MX. At a higher level of abstraction, it is worth considering whether diversity-related topics add additional factors not considered in *Factors of Modeller Experience*, such as cultural factors, or if they are simply moderators to the listed factors. In summary, we therefore propose the following.

Proposition 5 *In addition to the points raised in Proposition 1 to 4, the community needs to pay considerable attention to how collaboration and group diversity affects modelling.*

9 Conclusion

This paper summarises the discussions we, a group of 26 researchers and practitioners connected to MDE and human factors research, had during a one-week seminar at Schloss Dagstuhl. We focused on five topics which we elicited jointly

at the start of the seminar. In our opinion, these five topics require urgent attention in the MDE research community. We formulated research goals, questions, and propositions that support directing future initiatives towards an MDE community that is aware of and supportive of human factors and values.

The topics raised represent the collective opinions and knowledge of the various experts present at the seminar, and are often not evidence-based. Naturally, this means that others might disagree with our positions, and empirical evidence might prove us wrong in various ways in the future. We encourage the community to engage with us in constructive discussions on our positions and suggestions.

Finally, several suggested topics did not receive enough attention during the seminar. The list of goals, outcomes, and topics we presented in Sect. 2 of this report can serve as an inspiration to explore further directions in addition to the ones we proposed.

Appendix A: Goals and setup of the GI-Dagstuhl seminar

The results presented in this paper emerge from discussions at a GI-Dagstuhl seminar.⁵ These seminars are similar to the well-known traditional Dagstuhl seminars⁶ but cater towards early-career researchers, such as Ph.D. students and post-docs.

The primary goals of the seminar were (1) to establish closer collaborations and shape future goals in human factors in MDE and (2) to broaden the perspective of the seminar topic through insights into related research areas. To address (1), we invited early-career researchers who study human factors in software modelling through contacts and existing publications in the HuFaMo workshop series, held as a part of the models conference every year. Additionally, we invited a few selected senior researchers in this community. To address (2), we invited participants who conduct research not directly related to MDE but related to the human factors seminar theme, such as psychology or human-centred software engineering. Finally, connected to both goals, we invited experienced industrial participants interested in MDE, to anchor discussions in industrial needs.

A.1 Participants

The aforementioned goals led to a diverse group of 26 participants, with 11 women and 15 men, 3 participants from industry, another 2 who used to work in industry, 9 participants before finishing their Ph.D., and 5 senior participants.

⁵ <https://www.dagstuhl.de/en/seminars/gi-dagstuhl-seminars>.

⁶ <https://www.dagstuhl.de/en/seminars/dagstuhl-seminars>.

Given the nature of Dagstuhl seminars, most of the participants came from Germany (15), followed by Iceland (2), Portugal (2), and Sweden (2). One participant each came from Belgium, Canada, Switzerland, the Netherlands, and Spain.

A.2 Seminar structure

After a first unofficial get-together on Sunday, we started the seminar early on Monday, 19th November 2023. We ended after lunch on Friday, 24th November 2023, spending a total of 4.5 days on in-depth discussion. Throughout the workshop, we followed several of the practices recommended as a part of Liberating Structures,⁷ i.e. simple practices for structuring workshops and discussions engagingly and inclusively.

Monday was used to brainstorm seminar outcomes and topics. From Tuesday until Friday, we discussed the five most popular topics in breakout groups. Finally, we recorded results as a joint working document from Thursday onwards. In parallel to these activities, we had eight selected presentations from junior participants and one selected presentation on psychological theories from an established researcher.

A.3 Reflections on the seminar format

During one week, the author group discussed and reflected on human factors in MDE in a GI-Dagstuhl Seminar. The format we chose is akin to similar seminars, e.g. regular Dagstuhl Seminars, Lorentz Center Workshops,⁸ or NII Shonan Meetings.⁹ However, the seminar was aimed primarily towards early researchers, and we, on purpose, invited people with non-MDE backgrounds, i.e. conducting research in education and psychology. Additionally, we invited several practitioners with a background in MDE research. This diverse group composition led to many interesting and novel discussions, as participants were more likely to think “out of the box”, i.e. make connections to non-MDE research topics or industrial practice. Similarly, the juniority of the group as a whole might indeed have created more novel discussions, as many participants were not as mechanised in their thoughts on what MDE research is and should be. Hence, we believe the diverse group composition was a success and will consider this for future seminars with similar aims.

Acknowledgements We thank Schloss Dagstuhl for the opportunity to conduct the workshop and the GI (Gesellschaft für Informatik) for supporting it.

⁷ <https://www.liberatingstructures.com/>.

⁸ <https://www.lorentzcenter.nl/what-is-a-lorentz-center-workshop.html>.

⁹ <https://shonan.nii.ac.jp/proposal/guidelines/>.

References

- Jolak, R., et al.: Software engineering whispers: the effect of textual vs. graphical software design descriptions on software design communication. *Empir. Softw. Eng.* **25**, 4427–4471 (2020)
- Ho-Quang, T., Hebig, R., Robles, G., Chaudron, M.R., Fernandez, M.A. Juristo, N., Shepherd, D.: Practices and perceptions of uml use in open source projects. In: Juristo, N., Shepherd, D. (eds.) 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), pp. 203–212. IEEE (2017)
- Götz, S., Tichy, M., Groner, R.: Claimed advantages and disadvantages of (dedicated) model transformation languages: a systematic literature review. *Softw. Syst. Model.* **20**, 469–503 (2021)
- Liebel, G., Tichy, M., Knauss, E.: Use, potential, and showstoppers of models in automotive requirements engineering. *Softw. Syst. Model.* **18**, 2587–2607 (2019)
- Höppner, S., Haas, Y., Tichy, M., Juhnke, K.: Advantages and disadvantages of (dedicated) model transformation languages: a qualitative interview study. *Empir. Softw. Eng.* **27**, 159 (2022)
- France, R., Rumpe, B., Briand, L.C., Wolf, A.L.: Model-driven development of complex software: a research roadmap. In: Briand, L.C., Wolf, A.L. (eds.) *Future of Software Engineering (FOSE'07)*, pp. 37–54 (2007)
- Kalantari, R., Lethbridge, T.C.: Characterizing ux evaluation in software modeling tools: a literature review. *IEEE Access* **10**, 131509–131527 (2022)
- Evans, E.: *Domain-Driven Design Reference: Definitions and Pattern Summaries*. Dog Ear Publishing, Alaska (2014)
- González-Huerta, J., Insfrán, E., Abrahão, S., Moreira, A., Schätz, B., Gray, J., Vallecillo, A., Clarke, P.: Defining and validating a multimodel approach for product architecture derivation and improvement. In: Moreira, A., Schätz, B., Gray, J., Vallecillo, A., Clarke, P. (eds.) *Model-Driven Engineering Languages and Systems: 16th International Conference, MODELS 2013, Miami, FL, USA, September 29–October 4, 2013. Proceedings*, vol. 6, pp. 388–404. Springer (2013)
- Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R.: A taxonomy of tool-related issues affecting the adoption of model-driven engineering. *Softw. Syst. Model.* **16**, 313–331 (2015)
- Technical Committee ISO/TC 159/SC 4. ISO 9241-210:2019 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Tech. Rep., International Organization for Standardization (ISO) (2019). <https://www.iso.org/standard/77520.html>
- Mohagheghi, P., Gilani, W., Stefanescu, A., Fernandez, M.A.: An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases. *Empir. Softw. Eng.* **18**, 89–116 (2013)
- Pietron, J., Guerra, E., Iovino, L.: Enhancing collaborative modeling. In: Guerra, E., Iovino, L. (eds.) *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. Association for Computing Machinery* (2020)
- Ozkaya, M.: Are the uml modeling tools powerful enough for practitioners? A literature review. *IET Softw.* **13**, 338–354 (2019)
- Badreddin, O., Khandoker, R., Forward, A., Masmali, O., Lethbridge, T.C., Wasowski, A., Paige, R., Haugen, Ø.: A decade of software design and modeling: A survey to uncover trends of the practice. In: Wasowski, A., Paige, R., Haugen, Ø (eds.) *MODELS'18*, pp. 245–255. Association for Computing Machinery, New York, NY, USA (2018)
- Bordeleau, F., Liebel, G., Raschke, A., Stieglbauer, G., Tichy, M., Burgueño, L. et al.: Challenges and research directions for successfully applying mbe tools in practice. In: Burgueño, L. et al. (eds.) *Proceedings of MODELS 2017 Satellite Event: Workshops (ModComp, ME, EXE, COMMitMDE, MRT, MULTI, GEMOC, MoDeVva, MDETools, FlexMDE, MDEbug), Posters, Doctoral Symposium, Educator Symposium, ACM Student Research Competition, and Tools and Demonstrations co-located with ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS 2017), Austin, TX, USA, September, 17, 2017, Vol. 2019 of CEUR Workshop Proceedings*, pp. 338–343. CEUR-WS.org (2017). http://ceur-ws.org/Vol-2019/mdetools_1.pdf
- David, I., Aslam, K., Malavolta, I., Lago, P.: Collaborative model-driven software engineering—a systematic survey of practices and needs in industry. *J. Syst. Softw.* **199**, 111626 (2023)
- Franzago, M., Di Ruscio, D., Malavolta, I., Muccini, H.: Collaborative model-driven software engineering: a classification framework and a research map. *IEEE Trans. Softw. Eng.* **44**, 1146–1175 (2017)
- Hidayanto, A.N., Setyady, S.T.: Impact of collaborative tools utilization on group performance in university students. *Turk. Online J. Educ. Technol.* **13**, 88–98 (2014)
- Ur Rahman, A., Khan, K., Kamal, S.W., Naveed, H., Bacha, M.: Use of collaborative tools and modern technologies as critical success factor in global software development. *J. Softw. Eng.* **15**, 48 (2020)
- Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J., Dingel, J., Schulte, W., Ramos, I., Abrahão, S., Insfran, E.: Assessing the state-of-practice of model-based engineering in the embedded systems domain. In: Dingel, J., Schulte, W., Ramos, I., Abrahão, S., Insfran, E. (eds.), pp. 166–182 (2014)
- Exelmans, J., Pietron, J., Raschke, A., Vangheluwe, H., Tichy, M., Dubois, C., Cohen, J.: Optimistic versioning for conflict-tolerant collaborative blended modeling. In: Dubois, C., Cohen, J. (eds.) *FPVM 2022: 2nd International Workshop on Foundations and Practice of Visual Modeling, July 4–8, 2022, Nantes, France*, vol. 3250, pp. 1–12 (2022)
- Ryan, R., Deci, E.: Intrinsic and extrinsic motivations: classic definition and new directions. *Contemp. Educ. Psychol.* **25**, 54–67 (2000)
- Kuusinen, K., Petrie, H., Fagerholm, F., Mikkonen, T., Sharp, H., Hall, T.: Flow, intrinsic motivation, and developer experience in software engineering. In: Sharp, H., Hall, T. (eds.), pp. 104–117 (2016)
- Akdur, D., Garousi, V., Demirs, O.: A survey on modeling and model-driven engineering practices in the embedded software industry. *J. Syst. Archit.* **91**, 62–82 (2018)
- Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S., Taylor, R.N., Gall, H., Medvidovic, N.: Empirical assessment of MDE in industry. In: Taylor, R.N., Gall, H., Medvidovic, N. (eds.) *Proceedings of the 33rd International Conference on Software Engineering, ICSE'11*, pp. 471–480. Association for Computing Machinery, New York, NY, USA (2011)
- Vogelsang, A., Amorim, T., Pudlitz, F., Gersing, P., Philipps, J., Felderer, M. et al.: Should i stay or should i go? On forces that drive and prevent mbse adoption in the embedded systems industry. In: Felderer, M. et al. (eds.) *Product-Focused Software Process Improvement*, pp. 182–198. Springer, Cham (2017)
- Kalantari, R., Lethbridge, T., Blouin, A., Abrahão, S., Palanque, P., Selic, B.: Unveiling developers' mindset barriers to software modeling adoption. In: Blouin, A., Abrahão, S., Palanque, P., Selic, B. (eds.) *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (IEEE, 2023). Human Factors in Modeling workshop (HuFaMo)* (2023)
- Petre, M., Cheng, B.H.C., Pohl, K.: Uml in practice. In: Cheng, B.H.C., Pohl, K. (eds.) *Proceedings of the 2013 International Conference on Software Engineering, ICSE'13*, pp. 722–731 (IEEE) (2013)

30. Mussbacher, G., Dingel, J., Schulte, W., Ramos, I., Abrahão, S., Insfran, E. et al.: The relevance of model-driven engineering thirty years from now. In: Dingel, J., Schulte, W., Ramos, I., Abrahão, S., Insfran, E. (eds.) *Model-Driven Engineering Languages and Systems*, pp. 183–200. Springer (2014)
31. Hall, E.T.: *Beyond culture*. Anchor (1976)
32. Medel, P., Pournaghshband, V., Caspersen, M.E., Edwards, S.H., Barnes, T., Garcia, D.D.: Eliminating gender bias in computer science education materials. In: Caspersen, M.E., Edwards, S.H., Barnes, T., Garcia, D.D. (eds.) *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pp. 411–416 (2017)
33. Baltes, S., Diehl, S., Leavens, G.T., Garcia, A., Păsăreanu, C.S.: Towards a theory of software development expertise. In: Leavens, G.T., Garcia, A., Păsăreanu, C.S. (eds.) *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 187–200 (2018)
34. Catolino, G., Palomba, F., Tamburri, D.A., Serebrenik, A., Ferrucci, F., Kazman, R., Pasquale, L.: Gender diversity and women in software teams: How do they affect community smells? In: Kazman, R., Pasquale, L. (eds.) *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, 11–20 (IEEE, 2019)
35. Nunes, I., Moreira, A., Araujo, J.: Gire: Gender-inclusive requirements engineering. *Data Knowl. Eng.* **143**, 102108 (2023)
36. Vrieler, T., Nylén, A., Cajander, Å.: Computer science club for girls and boys—a survey study on gender differences. *Comput. Sci. Educ.* **31**, 431–461 (2021)
37. Baron-Cohen, S., Belmonte, M.K.: Autism: a window onto the development of the social and the analytic brain. *Annu. Rev. Neurosci.* **28**, 109–126 (2005)
38. David, I., et al.: Blended modeling in commercial and open-source model-driven software engineering tools: a systematic study. *Softw. Syst. Model.* **22**, 415–447 (2023)
39. Schwartz, S.H.: An overview of the Schwartz theory of basic values. *Online Read. Psychol. Cult.* **2**, 11 (2012)
40. Stikkolorum, D.R.: *Studies into interactive didactic approaches for learning software design using UML*. Ph.D. thesis, Leiden University (2022)
41. Easterbrook, S., Lethbridge, T., Cabot, J., Egyed, A.: Modelling the climate system: Is model-based science like model-based engineering?(keynote). In: Lethbridge, T., Cabot, J., Egyed, A. (eds.) *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 1–1. IEEE (2015)
42. Blair, G., Baudry, B., Combemale, B., Kienzle, J., Pretschner, A.: Grand challenges, grand responses?(keynote). In: Baudry, B., Combemale, B., Kienzle, J., Pretschner, A. (eds.) *2016 ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS)*. ACM (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Grischa Liebel is an Assistant Professor in Software Engineering and the director of the CRESS research centre at Reykjavik University, Iceland. He holds a Ph.D. degree from Chalmers University of Technology, Sweden. His research focuses on human factors in software engineering, typically in areas such modelling and model-based engineering, requirements engineering, processes, and education. Much of Grischa's research is done in collaboration with industry (and with humans).

Jil Klünder is a post-doctoral researcher and lecturer at Leibniz University Hannover. She studied Mathematics and did her Ph.D. in 2019 in software engineering. In her Ph.D. thesis, she analysed social aspects such as communication and interactions in meetings in collaboration with psychologists. Afterwards, her research mainly focused and still focuses on socio-technical software engineering, including sentiment analysis, and on software processes.

Regina Hebig is a Professor of Software Engineering at the University of Rostock, Germany. She did her Ph.D. at the University of Potsdam, Germany, in 2014 and her docent degree at the University of Gothenburg in 2019, where she worked as an Associate Professor until her change to Rostock in 2023. Her research interests include software evolution, software modelling, and software processes.

Christopher Lazik is a Ph.D. candidate and researcher in Software Engineering and Human–Computer Interaction at Humboldt University in Berlin. His research focuses on designing software systems that address the challenges associated with human participation.

Inês Nunes is currently a Graduate Programme Participant at the European Central Bank. She completed her M.Sc. in Computer Science in 2021 from NOVA University of Lisbon, School of Science and Technology, where she researched the topic of gender inclusion in Requirements Engineering.

Isabella Graßl is a Ph.D. student at the Chair of Software Engineering II at University of Passau, Germany, and interested in interdisciplinary research on social and collaborative aspects of software development. In her dissertation, she is investigating the effects of thematic preference and group constellation in programming education with regard to various aspects of diversity, in particular with the aim of inspiring female students for computer science.

Jan-Philipp Steghöfer is a Senior Researcher at XITASO. His research interests include security assurance, security of AI systems, agile development in regulated domains, and model-driven engineering. Jan-Philipp is an active member of the software engineering community where he is part of the organisation committee for events like the IEEE International Requirements Engineering Conference (RE). He also reviews for ICSE, TSE, JSS, ESEM, and many other venues.

Joeri Exelmans is a Ph.D. student at the Modeling, Simulation, and Design (MSDL) research laboratory at the University of Antwerp, Belgium. The topic of his Ph.D. is the modular composition of modelling languages and their (modelling, simulation, and debugging) tools.

Julian Oertel studied computer science at the University of Rostock and completed his master's degree with a thesis on planning heterogeneous data traffic in time-sensitive networks. He is now pursuing a Ph.D. at the Chair of Software Engineering, where relevant research areas include Software Comprehension, Code Summarization, and Human Factors in Software Development.

Kai Marquardt graduated from the Karlsruhe Institute of Technology (KIT) with degrees in Geography, Mathematics, and Informatics to become a teacher. Currently, he is a doctoral researcher at the Faculty for Informatics at the KIT. His research interests focus on the social aspects of computer science and digitalisation in computing education.

Katharina Juhnke is a project manager for software automation and value-stream responsible for optics at Carl Zeiss SMT GmbH. Her research expertise is in engineering model-based software solutions with high industrial impact. You can contact the author at katharina.juhnke@zeiss.com

Kurt Schneider is a professor of software engineering at Leibniz Universität Hannover. He holds a doctoral degree from Stuttgart University (Germany). From 1994–1996, he had a postdoctoral position at the Center for LifeLong Learning and Design (L3D) at the University of Colorado at Boulder. He was researcher and manager at the Daimler Research Center in Ulm for eight years. His research interests focus on the intersection of humans with software technology, such as in requirements engineering, usability engineering, or explainability.

Lucas Gren has a Ph.D. degree in Software Engineering and is a Senior Lecturer at Chalmers and the University of Gothenburg. He also has master's degrees in software engineering, Social Psychology, and Business Administration Management. Today Lucas does research within the interdisciplinary field of combining Organizational Psychology with Software and AI Engineering. He also works as a consultant and helps a diversity of companies both nationally and internationally.

Lucia Happe is a Senior Research Fellow with the Faculty of Informatics and the Center for Teacher Education at the Karlsruhe Institute of Technology, based in Karlsruhe, Germany. Dr. Happe's research is centred on model-driven software engineering, interdisciplinary computing education, and the application of emerging learning technologies. She is an expert in performance engineering and is dedicated to improving the quality of models, metamodels, and model transformations. Dr. Happe also founded RockStartIT, a pioneering non-profit that revitalises computing education to engage and include a broad spectrum of participants. To discuss potential collaborations or to learn more about her work, Dr. Happe can be reached through her LinkedIn at <https://www.linkedin.com/in/lucia-happe/> or by email at lucia.happe@kit.edu.

Marc Herrmann is currently working towards the Ph.D. degree in computer science with the Leibniz University Hannover. His research focuses on socio-technical aspects of software engineering, such as the application of sentiment analysis within software development environments. He was lead organiser of the Fifth International Workshop on Affective Computing in Requirements Engineering.

Marvin Wyrich is a researcher at the Saarland University, where he has been part of the Chair of Software Engineering since 2023. His research interests include empirical and behavioural software engineering, with a focus on program comprehension, science communication, and developing sound research methodologies.

Matthias Tichy is a full professor for software engineering at Ulm University and director of the Institute of Software Engineering and Programming languages. His research focuses on model-driven software engineering, particularly, for cyber-physical systems, and empirical research, particularly, developer experience. You can contact the author at matthias.tichy@uni-ulm.de.

Miguel Goulão is an Associate Professor at the NOVA University of Lisbon. He is an integrated member of the Software Systems group of NOVA LINCS. He has a Ph.D. (2008) in Informatics from Universidade Nova de Lisboa. The broad aim of his research is to improve the software developer's experience, to better deal with software development complexity. Miguel uses Evidence-Based, Empirical Software Engineering, and User Experience evaluation techniques to identify the strengths and shortcomings in languages, tools, and approaches. He is particularly interested in improving the understandability of Requirements Engineering and Domain-Specific Languages. Miguel is also active in research on Equity, Diversity and Inclusion in Software Engineering, and is a working group leader in the EUGAIN COST Action. Miguel published over 70 papers in peer-reviewed journals and conferences and served as guest editor of special issues in international journals, as a PC member, and as PC and Organising Chair in several events.

Rebekka Wohlrab is an assistant professor at Chalmers University of Technology and an adjunct faculty at Carnegie Mellon University. Her research interests include requirements engineering and software architecture, especially in connection with human-in-the-loop self-adaptive systems. Wohlrab received her Ph.D. in computer science from Chalmers University of Technology.

Reyhaneh Kalantari received her B.S. degree in 2012 and followed it with an M.S. in IT Management in 2016 from TMU University in Iran. Currently, she is pursuing her doctoral studies, specialising in digital transformation and innovation at the University of Ottawa, Canada. Since 2020, Reyhaneh has held pivotal roles as a Research and Teaching Assistant, alongside serving as a part-time professor at the University of Ottawa. Her doctoral research has been focused on advancing user experience within software modelling tools. Reyhaneh's scholarly pursuits reflect her commitment to bridging the gap between technology and user needs, aiming to shape more intuitive and effective digital solutions.

Robert Heinrich heads the Quality-driven System Evolution research group at Karlsruhe Institute of Technology (KIT). His research interests include Secure Software Systems, Model-driven Engineering, and Software Evolution. He is involved in the organisation committees of several international conferences, initiated and organised various research seminars and workshops, and is reviewer for international premium journals.

Sandra Greiner is Assistant Professor in the Artificial Intelligence, Cybersecurity, and Programming Languages Section at the Southern University of Denmark. Before, she was postdoctoral researcher at the University of Bern from Nov. 2022 until April 2024. Sandra earned her Ph.D. in Software Engineering at the University of Bayreuth and studied Computer Science (M.Sc.) and its application with Engineering Sciences (B.Sc.) there. In 2021, she visited the IT University of Copenhagen, granted as a scholarship by the German Academic Exchange Service (DAAD), to investigate synchronising variability information in collaborative development. Her research interests include model-driven software-engineering, software product line engineering, software evolution, and adopting software engineering to enhance industrial applications. She regularly co-organises community workshops and strives to systematically and consistently manage software-intensive systems over time.

Satrio Adi Rukmono is a Ph.D. candidate in the Software Engineering and Technology group at Eindhoven University of Technology (TU/e), the Netherlands. He holds master's and bachelor's degrees in informatics from Institut Teknologi Bandung (ITB), Indonesia. His research focuses on bridging the gap between software architecture and its implementation, including the use of artificial intelligence for reconstructing software architectures and generating explanations thereof. Satrio has experience teaching programming and software engineering courses at both ITB and TU/e.

Shalini Chakraborty completed her bachelor's (2017) and master's (2019) in Computer Science from Calcutta University, India. She is pursuing her Ph.D., working in Software Engineering (SE), primarily focusing on Model-Based Engineering (MBE) and empirical research. She is interested in human factors in SE and social software engineering. Currently, she is employed at Reykjavik University as a Ph.D. candidate and teaching assistant.

Silvia Abrahão is Full Professor of Software Engineering in the Department of Computer Science at the Universitat Politècnica de València (Spain). She is also Director of the Master's Program in Software Systems Engineering and Technology at UPV. Her research interests include the application of machine learning techniques to support the adaptation of user interfaces, quality assurance in model-driven engineering, the empirical assessment of software modelling approaches, and the integration of usability/UX into software development. Contact her at <https://sabrahao.wixsite.com/dsic-upv> or sabrahao@dsic.upv.es.

Vasco Amaral is an Associate Professor at the Department of Informatics at NOVA School of Sciences and Technology of NOVA University of Lisbon since 2005, President of the Informatics Engineering College at the Portuguese Order of Engineers (Ordem of Engenheiros), IFIP Portuguese representative (2022–2025), and Has a Ph.D. from Mannheim University in 2005. His major areas of interest are Software Engineering, Model-Driven Software Development, Domain-Specific Languages, and Cyber-Physical Systems Modelling. He has worked as a research collaborator at CERN (Geneva) and DESY (Hamburg) as a LIP-Lisbon member. He is currently a research-integrated member at NOVA LINCS. He was vice-chair of a COST Action on Multi-Paradigm Modelling for Cyber-physical Systems. He has also organised a series of 8 international doctoral Summer Schools (DSM-TP), a doctoral training school (MPM4CPS 2018), and workshops (MPM, HuFaMo). He was co-editor of Journal Special issues (Software Quality Journal, Comput. Lang. Syst. Struct.). He was also involved in organising international conferences (co-chair and PC chair of QUATIC14 and 2016, and other organisation roles in conferences such as VL/HCCC and MODELS). He supervised 2 Postdocs, 4 Ph.D. students, 3 Ph.D. ongoing supervision, and >70 M.Sc. students. He has >100 peer-reviewed publications in journals, conferences and workshops.

Authors and Affiliations

Grischa Liebel¹ · Jil Klünder² · Regina Hebig³ · Christopher Lazik⁴ · Inês Nunes⁵ · Isabella Graßl⁶ · Jan-Philipp Steghöfer⁷ · Joeri Exelmans⁸ · Julian Oertel³ · Kai Marquardt⁹ · Katharina Juhnke¹⁰ · Kurt Schneider² · Lucas Gren¹¹ · Lucia Happe¹² · Marc Herrmann² · Marvin Wyrich¹³ · Matthias Tichy¹⁴ · Miguel Goulão¹⁵ · Rebekka Wohlrab¹¹ · Reyhaneh Kalantari¹⁶ · Robert Heinrich¹⁷ · Sandra Greiner¹⁸ · Satrio Adi Rukmono¹⁹ · Shalini Chakraborty¹ · Silvia Abrahão²⁰ · Vasco Amaral¹⁵

✉ Grischa Liebel
grischal@ru.is

✉ Jil Klünder
jil.kluender@inf.uni-hannover.de

✉ Regina Hebig
regina.hebig@uni-rostock.de

Christopher Lazik
lazikchr@informatik.hu-berlin.de

Inês Nunes
ir.nunes@campus.fct.unl.pt

Isabella Graßl
isabella.grassl@uni-passau.de

Jan-Philipp Steghöfer
jan-philipp.steghoefer@xitaso.com

Joeri Exelmans
joeri.exelmans@uantwerpen.be

Julian Oertel
julian.oertel@uni-rostock.de

Kai Marquardt
kai.marquardt@kit.edu

Katharina Juhnke
katharina.juhnke@zeiss.com

Kurt Schneider
kurt.schneider@inf.uni-hannover.de

Lucas Gren
lucas.gren@gu.se

Lucia Happe
lucia.happe@kit.edu

Marc Herrmann
marc.herrmann@inf.uni-hannover.de

Marvin Wyrich
wyrich@cs.uni-saarland.de

Matthias Tichy
matthias.tichy@uni-ulm.de

Miguel Goulão
mgoul@fct.unl.pt

Rebekka Wohlrab
wohrlab@chalmers.se

Reyhaneh Kalantari
Reyhaneh.kalantari@uottawa.ca

Robert Heinrich
robert.heinrich@kit.edu

Sandra Greiner
sandra.greiner@unibe.ch

Satrio Adi Rukmono
s.a.rukmono@tue.nl

Shalini Chakraborty
shalini19@ru.is

Silvia Abrahão
sabraha@dsic.upv.es

Vasco Amaral
vasco.amaral@fct.unl.pt

¹ Department of Computer Science, Reykjavik University, Menntavegur 1, 102 Reykjavík, Iceland

² Software Engineering Group, Leibniz University Hannover, Welfengarten 1, 30167 Hannover, Germany

³ Department of Software Engineering, University of Rostock, Albert-Einstein-Straße, 18057 Rostock, Germany

⁴ Department of Software Engineering, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany

⁵ Unaffiliated Researcher, Frankfurt, Germany

⁶ Department of Software Engineering II, University of Passau, Innstraße 33, 94032 Passau, Germany

⁷ XITASO GmbH, Austraße 35, 86153 Augsburg, Germany

⁸ Department of Computer Science, University of Antwerp, Middelheimlaan 1, 2020 Antwerp, Belgium

⁹ Department of Informatics, Karlsruhe Institute of Technology, Am Fasanengarten 5, 75131 Karlsruhe, Germany

¹⁰ SMT-EMI6, Carl Zeiss SMT GmbH, Rudolf-Eber-Straße 2, 73447 Oberkochen, Germany

¹¹ Department of Computer Science and Engineering, Chalmers | University of Gothenburg, 411 27 Gothenburg, Sweden

¹² Department of Interdisciplinary Didactics, Karlsruhe Institute of Technology, Engesserstr. 2, 75131 Karlsruhe, Germany

¹³ Department of Computer Science, Saarland University, Saarland Informatics Campus, 66123 Saarbrücken, Germany

¹⁴ Institute of Software Engineering and Languages, Ulm University, James-Franck-Ring, 89081 Ulm, Germany

¹⁵ Department of Computer Science, NOVA School of Science & Technology, Campus de Caparica, 2829-516 Caparica, Portugal

¹⁶ Department of Engineering Design and Teaching Innovation, University of Ottawa, 800 King Edward Avenue, Ottawa K1N 6N5, Canada

¹⁷ KASTEL - Institute of Information Security and Dependability, Karlsruhe Institute of Technology, Am Fasanengarten 5, 75131 Karlsruhe, Germany

- ¹⁸ Department of Computer Science, University of Bern, 3012 Bern, Switzerland
- ¹⁹ Department of Mathematics and Computer Science, Eindhoven University of Technology, De Zaale, 5600 MB Eindhoven, The Netherlands

- ²⁰ Instituto Universitario Mixto de Tecnología Informática, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain