Information and Software Technology

# Understanding the Landscape of Software Modelling Assistants: A Systematic Mapping

--Manuscript Draft--

| | |
|---|---|
| **Manuscript Number:** | INFSOF-D-23-00931 |
| **Article Type:** | Review article |
| **Keywords:** | Modelling assistance; Model-driven development; Systematic mapping; State of the practice; Low-code; No-code |
| **Abstract:** | Abstract. Context. Model Driven Software Engineering (MDSE) and low-code/no-code software development tools promise to increase quality and productivity by modelling instead of coding software. One of the major advantages of modelling software is the increased possibility to involve diverse stakeholders since it removes the barrier of being IT experts to actively participate in software production processes. From an academic and industry point of view the main question remains: What has been proposed to assist humans in software modelling tasks? Objective. In this paper, we systematically elucidate the state of the art in assistants for software modelling and their use in MDSE and low-code/no-code tools. Method. We conducted a systematic mapping to review the state of the art and answer the following research questions: i) how is software modelling assisted? ii) what goals and limitations do existing MDSE tools report in terms of modelling assistance? iii) which evaluation metrics and target users do existing MDSE tools consider for modelling assistance? For this purpose, we selected 44 proposals from 2.613 screened records and reviewed 17 MDSE and low-code/no-code tools from main market players published by the Gartner Magic Quadrant. Result. We clustered existing proposals regarding their modelling assistance strategies, goals, limitations, evaluation metrics, and target users, both in research and practice. Conclusions. We found that both academic and industry proposals recognise the value of assisting software modelling. However, documentation about MDSE assistants' limitations, evaluation metrics, and target users is scarce or non-existent. With the advent of artificial intelligence, we expect more assistants for MDSE and low-code/no-code software development will emerge, making imperative the need for well-founded frameworks for designing modelling assistants focused on addressing target users' needs and advancing the state of the art. We compiled our results within an emerging framework to support the design of MDSE modelling assistants. |

Structured Abstract

**Abstract. Context.** Model Driven Software Engineering (MDSE) and low-code/no-code software development tools promise to increase quality and productivity by modelling instead of coding software. One of the major advantages of modelling software is the increased possibility to involve diverse stakeholders since it removes the barrier of being IT experts to actively participate in software production processes. From an academic and industry point of view the main question remains: What has been proposed to assist humans in software modelling tasks? **Objective.** In this paper, we systematically elucidate the state of the art in assistants for software modelling and their use in MDSE and low-code/no-code tools. **Method.** We conducted a systematic mapping to review the state of the art and answer the following research questions: i) how is software modelling assisted? ii) what goals and limitations do existing MDSE tools report in terms of modelling assistance? iii) which evaluation metrics and target users do existing MDSE tools consider for modelling assistance? For this purpose, we selected 44 proposals from 2.613 screened records and reviewed 17 MDSE and low-code/no-code tools from main market players published by the Gartner Magic Quadrant. **Result.** We clustered existing proposals regarding their modelling assistance strategies, goals, limitations, evaluation metrics, and target users, both in research and practice. **Conclusions.** We found that both academic and industry proposals recognise the value of assisting software modelling. However, documentation about MDSE assistants' limitations, evaluation metrics, and target users is scarce or non-existent. With the advent of artificial intelligence, we expect more assistants for MDSE and low-code/no-code software development will emerge, making imperative the need for well-founded frameworks for designing modelling assistants focused on addressing target users' needs and advancing the state of the art. We compiled our results within an emerging framework to support the design of MDSE modelling assistants.

# Understanding the Landscape of Software Modelling Assistants: A Systematic Mapping*

David Mosquera (corresponding author) [1[0000-0002-0552-7878]], Marcela Ruiz[1[0000-0002-0592-1779]], Oscar Pastor[2[0000-0002-1320-8471]], and Jürgen Spielberger[1[0000-0003-2617-3535]]

[1] Zürich University of Applied Sciences, Gertrudstrasse 15, Winterthur 8400, Switzerland
{mosq, ruiz, spij}@zhaw.ch
[2] PROS-VRAIN: Valencian Research Institute for Artificial Intelligence – Universitat Politècnica de València, Camí de Vera, s/n, València 46022, Spain
opastor@dsic.upv.es

**Abstract. Context.** Model Driven Software Engineering (MDSE) and low-code/no-code software development tools promise to increase quality and productivity by modelling instead of coding software. One of the major advantages of modelling software is the increased possibility to involve diverse stakeholders since it removes the barrier of being IT experts to actively participate in software production processes. From an academic and industry point of view the main question remains: What has been proposed to assist humans in software modelling tasks? **Objective.** In this paper, we systematically elucidate the state of the art in assistants for software modelling and their use in MDSE and low-code/no-code tools. **Method.** We conducted a systematic mapping to review the state of the art and answer the following research questions: i) how is software modelling assisted? ii) what goals and limitations do existing MDSE tools report in terms of modelling assistance? iii) which evaluation metrics and target users do existing MDSE tools consider for modelling assistance? For this purpose, we selected 44 proposals from 2.613 screened records and reviewed 17 MDSE and low-code/no-code tools from main market players published by the Gartner Magic Quadrant. **Result.** We clustered existing proposals regarding their modelling assistance strategies, goals, limitations, evaluation metrics, and target users, both in research and practice. **Conclusions.** We found that both academic and industry proposals recognise the value of assisting software modelling. However, documentation about MDSE assistants' limitations, evaluation metrics, and target users is scarce or non-existent. With the advent of artificial intelligence, we expect more assistants for MDSE and low-code/no-code software development will emerge, making imperative the need for well-founded frameworks for designing modelling assistants focused on addressing target users' needs and advancing the state of the art. We compiled our results within an emerging framework to support the design of MDSE modelling assistants.

**Keywords:** Modelling assistance, Model-driven development, Systematic mapping, State of the practice, Low code, No-code.

* **Abbreviations**. MDSE: Model Driven Software Engineering; IDE: Integrated Development Environments; MRQ: Main Research Question; GMQ: Gartner Magic Quadrant; RQ: Research question; I: Inclusion criteria; E: Exclusion criteria; G: Goal; L: Limitation; NS: Not specified; MDE: Model Driven Engineering; M: Evaluation Metric; U: Target User; NE: Not Evaluated; LE: Leaders; C: Challengers; V: Visionaries; NP: Niche Players; NF: Not found.

# 1 Introduction

Model Driven Software Engineering (MDSE) and low-code/no-code tools are software development tools that uses models as the main input to generate software. That includes model-driven development tools, model-based development tools, low-code tools, and no-code tools, among others. MDSE tools aim to increase software development teams' productivity and decrease software time-to-market [1]. However, software modelling tools are not yet widely adopted in practice compared to Integrated Development Environments (IDE) for coding. This lack of adoption motivated different empirical research efforts to discover the benefits of such tools. Results [2, 3] show that MDSE tools are still maturing and improving modelling assistance is an identified not-yet-solved challenge [4]. Understanding the behaviour and skills of users, understanding the modelling context, and increasing the automation scope are examples of not yet solved challenges in modelling assistance.

***Modelling assistance:*** we refer to "modelling assistance" in this paper as any software artefact, method, or technique that aims to assist users during software modelling tasks in the context of an MDSE tool—i.e., to assist users in developing software using models in an MDSE tool.

Thus, improving modelling assistance is a contemporary challenge in information systems engineering, calling for new solutions. However, how can we improve without knowing what is the state of development in this area? Other authors have made the same question and motivate them to perform literature reviews on specific domains, technologies, and strategies [5–11]. Nevertheless, to the best of our knowledge, there is a lack of a review both in literature and practice without any restriction regarding technology, strategy, or domain. Therefore, in this paper, we propose the following main research question (MRQ):

***MRQ****: What proposals exist in the literature and practice to assist humans during modelling tasks in MDSE tools?*

From the perspective of literature, we conducted a systematic mapping study based on Petersen et al. [12] guidelines to gather data from the literature around our MRQ. We extracted data around i) how is software modelling assisted? ii) what goals and limitations do existing MDSE tools report in terms of modelling assistance? and iii) which evaluation metrics and target users do existing MDSE tools consider for modelling assistance?

To do so, we implemented two search strategies: database search and snowballing. We selected five scientific databases for the database search and performed the search using a search string based on PICO (Population, Intervention, Comparison and Outcomes) keywords [13]. We selected an initial set of papers based on the database search for snowballing and then performed forward and backward snowballing. Three reviewers selected 44 proposals from a set of 2.613 records using inclusion/exclusion criteria. We evaluated the inclusion reliability using the K-statistic [14], resulting in a substantial inclusion agreement [15]—i.e., having a $0.8 > K\text{-statistic} > 0.61$. Having 44 proposals selected, we extracted the data around the MRQ. We clustered, analysed, and discussed the data.

***Strategy:*** we refer to "strategy" in this paper as how authors propose to achieve a goal—for example, a proposal aiming to assist users in creating models by using a recommender system; here, the recommender system is the strategy.

We observed that some proposals mainly assist users during modelling tasks in MDSE tools using software-based strategies, including tools, techniques, methods, frameworks, and languages. In contrast, other proposals use no-software-based strategies such as guidelines. Regarding goals, we found seven clusters of goals such as addressing change propagation, enhancing consistency checking, ensuring model compatibility, improving model quality, improving user interaction, easing model evolution and supporting vulnerability detection. Thus, we observed that proposals aim to assist users in creating and refining existing models in MDSE tools. On the other hand, we identified five limitation clusters, including accuracy, effort, generality, learnability, and scope limitations. Similarly, we found three clusters for evaluation metrics based on the Technology Acceptance Model (TAM) [16], including effectiveness, efficiency, and user perception metrics. Regarding target users, we found three clusters: designers/modellers, domain experts, and software developers. However, we observed that several proposals have no specified limitations, evaluation metrics, and target users, hindering gap identification, further analysis, and arising inconsistencies.

From the perspective of practice, we reviewed 17 MDSE tools from the Gartner Magic Quadrant for Enterprise Low-Code application platforms [17]. We extracted data about: what is the state of the practice on modelling assistance? We explored the tools' documentation and extracted quotes about strategies, goals, limitations, target users, and metrics documented in practice for assisting during modelling tasks. We observed that some MDSE tools have more than one strategy for assisting during users modelling tasks. However, not all MDSE tools have documented their modelling assistants, hindering data extraction and comparison.

After analysing our results, we have provided an outlook on the modelling assistance research area. We provide a summary on our results from a strategic point of view, considering the goals, limitations, and evaluation metrics we found. Our research highlights a gap in software modelling assistance, especially in terms of explicitly defining limitations, evaluation metrics, and target users. Moreover, we anticipate disruptive changes in the modelling assistance strategies and goals with the rise of artificial intelligence-driven technologies in the following years. To face this disruption and close the identified gap, we consider proposing a shared framework essential for improving software modelling assistants' design. Thus, based on our results, we introduce AssistMDSE: an emerging framework that considers the who, how, and what for assisting MDSE users during modelling tasks. We exemplify the use of AssistMDSE and the results of our study on an illustrative case, designing a modelling assistant for automatically creating models in the digital health domain.

We expect researchers and MDSE practitioners to benefit from our results and emerging framework to design such new software modelling assistants, relying on existing goals and targeting existing gaps, avoid mentioned limitations, consider listed evaluation metrics, and target specific users. As a final step, we'll discuss how our study aligns with our research agenda, which includes the creation of new resources for the

community, such as a public repository for storing modelling assistance proposals. We provide a complete overview of our research in Fig. 1.

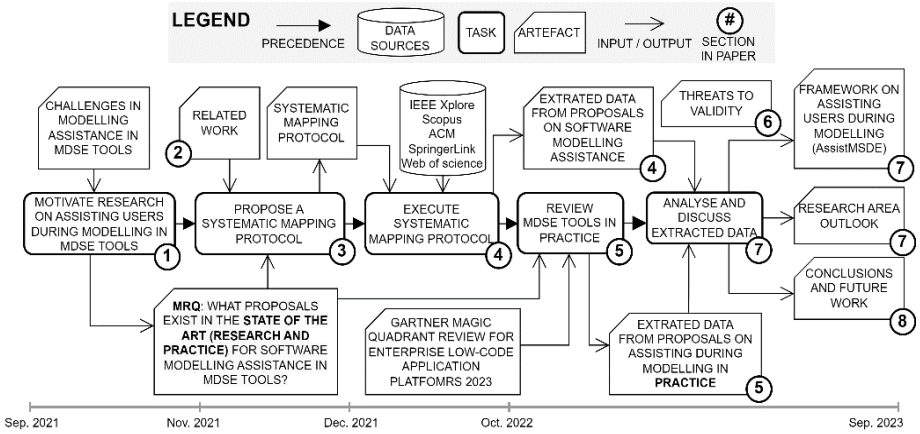



**Fig. 1.** Research overview.

This paper is structured as follows: In Section 2, we review the related works on systematic mappings about software modelling assistance; In Section 3, we show the systematic mapping study protocol; In Section 4, we explore the results from executing the systematic mapping study protocol; In Section 5, we conduct the review on MDSE tools available in practice based on the GMQ; In Section 6, we discuss the threats to validity and limitations of our study; In Section 7, we provide an outlook using the obtained results from both literature and practice, provide insights, and propose an emerging framework for designing proposals on assisting during modelling tasks (AssistMDSE); and, finally, in Section 8, we conclude our paper and point out future work.

## 2 Related work

In the literature, there are various systematic mapping and literature review studies that we grouped into two major groups: i) modelling assistance for Model-Driven Engineering tools (MDE)—i.e., proposals to assist in creating MDSE tools using an MDE tool like the Eclipse Modelling Framework [18]—and ii) specific strategies to assist users during modelling tasks in MDSE tools—e.g., reviewing proposals that use recommender systems as a strategy to assist users during modelling tasks in MDSE tools. We describe in detail such related works in the following paragraphs.

- **Reviews on modelling assistance for MDE tools.** The first group of related works aims to gather proposals around assisting modelling tasks from the model-driven engineering perspective. These studies review techniques, tools, and methods to assist model-driven engineers in creating MDSE tools, limiting the scope to MDSE engineers rather than MDSE tools' users. For instance, a systematic literature review on developing model transformations [5], a systematic mapping study on domain-

specific language development tools [6, 7], and a systematic literature review on testing model-to-text transformations [8].

- **Reviews on modelling assistance for MDSE tools using specific strategies.** The second group of related works aims to review proposals to assist users during modelling tasks in MDSE tools—i.e., proposals that assist MDSE tools' users in modelling software. These related works gather proposals with a common strategy to assist modelling tasks. Although this information is useful for understanding the existing proposals following a specific strategy, their analysis is restricted lacking generality. For instance, a systematic mapping review on recommender systems in MDSE tools [9], a systematic literature review on elicitation techniques in MDSE [10], and a research map on collaborative MDSE [11].

To the best of our knowledge, we lack a review on modelling assistance literature and practice in MDSE tools without any restriction regarding a specific strategy. This gap motivates our study.

## 3 Systematic Mapping Study Design[1]

Systematic mapping studies are designed to overview a research area, searching the literature to know what topics have been covered [12]]. We consider the following steps in our systematic mapping study: research questions, search strategy, inclusion/exclusion criteria, quality assessment, and data extraction. We deeply explore each step in the following subsections. Moreover, we show graphically each step, including inputs and outputs, in Fig. 2.

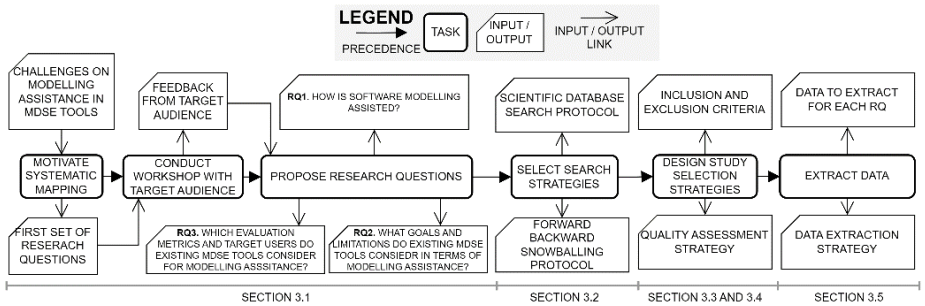



**Fig. 2.** Systematic mapping study design overview.

### 3.1 Research questions

In Section 1, we introduced the MRQ of our study: *What proposals exist in the literature and practice to assist humans during modelling tasks in MDSE tools?* This MRQ

[1] To facilitate further replication of this systematic mapping study, all material related to the protocol, research questions, search strategy, inclusion/exclusion criteria, quality assessment, data extraction, triangulation, and results can be found at https://zenodo.org/records/10262145.

is a general statement that motivates our study. Since this MRQ is highly generic, we need to split the MRQ into a set of more-specific research questions (RQs), related to a literature review. We propose a set of RQs based on our experience with the topic. Based on Petersen et al. [12] guidelines, consulting the proposed RQs with the target audience is a good practice. Thus, we bring together nine software engineering experts from the Software Engineering research group at the Zurich University of Applied Sciences (ZHAW). We consider them viable subjects to answer as target audience of our study since they work on software-modelling-related topics in the research group. Based on the initial set of RQs and MRQ, we ask them: i) *which RQ you found more critical for dividing the MRQ?* and ii) *what other RQ would you propose to divide the MRQ?* After discussing the results, we divide the MRQ into three RQs:

- **RQ1.** *How is software modelling assisted?* Answering this question, we aim to gather knowledge about the different strategies that have been developed to assist during modelling tasks in MDSE tools. We expect to gather a set of tools, methods, techniques, and frameworks, among others that have been proposed for assisting during modelling tasks. This can help researchers, users, and developers to expand their understanding of modelling assistant proposals and keep up with the last advancements in the field.
- **RQ2.** *What goals and limitations do existing MDSE tools report in terms of modelling assistance?* Proposals' goals and limitations give a deeper understanding of their potential impact and usefulness to be implemented in a specific context. We expect to gather a set of goals and limitations that allows researchers, users, and developers to make informed judgments about the viability of a specific proposal in their contexts.
- **RQ3.** *Which evaluation metrics and target users do existing MDSE tools consider for modelling assistance?* Evaluation metrics and target users are relevant for understanding the proposals' maturity. Identifying evaluation metrics, researchers, users, and developers can manage their expectations on how successful a proposal could be based on empirical results. Moreover, knowing the target users, researchers, users, and developers can better assess a specific group or communities using a specific proposal.

### 3.2 Search strategy

In our systematic mapping study, we combine two search strategies: database search and snowballing. First, we perform a database search in five scientific databases: IEEE Xplore, ACM Digital Library, Scopus, Springer Link, and Web of Science. We require a search string to search in such scientific databases. Therefore, we use the PICO (Population, Intervention, Comparison and Outcomes) [13] to identify the keywords and formulate the search string as follows:

- **Set of terms comprising the Population**. In our study, we consider population as MDSE tools. We exclude generic modelling tools—e.g., draw.io, diagrams.net, lucidchart.com—since our goal is analysing modelling assistants in tools that automatically produce software as an output. Therefore, we propose the following terms for

describing our population: *Model-Driven Development (MDD), Model-Driven Engineering (MDE), Model-Driven Architecture (MDA), Model Driven Software Engineering (MDSE), Model-Based Software Engineering (MBSE), Low code, No-code.*

- **Set of terms related to the Intervention.** In our study, we focus on proposals that aim to assist "users" during modelling tasks as intervention. We exclude any term related to specific strategy or tool to assist such "users" during modelling tasks—e.g., recommender system, machine learning, rule-based assistant—since our goal is to gather as many proposals as possible no matter the strategy. Moreover, we aim to include specific terms about the "user" as the intervention since we aim to gather proposals that have a human user. Therefore, we propose the following terms for representing our intervention: *assist, support, help, maintain, promote, ease, facilitate, simplify, user, developer, tester, software engineer, architect, usability, usage, approach, proposal, concept, idea, method, manner, technique, procedure, program, assistant.*

Then, we propose the following search string based on such a set of keywords:

***Search string:*** *("Model-Driven Engineering"* **OR** *"MDE"* **OR** *"Model-Driven Architecture"* **OR** *"MDA"* **OR** *"Model Driven Software Engineering"* **OR** *"MDSE"* **OR** *"Model-Driven Development"* **OR** *"MDD"* **OR** *"Model-Based Software Engineering"* **OR** *"MBSE"* **OR** *"Low code" OR "No code")* **AND** *("support*"* **OR** *"assist*"* **OR** *"help*"* **OR** *"ease"* **OR** *"facilitate*"* **OR** *"simplify*")* **NEAR TO** *("user"* **OR** *"developer"* **OR** *"tester"* **OR** *"software engineer"* **OR** *"analyst"* **OR** *"architect" OR "usability" OR "usage")* **AND** *("approach"* **OR** *"proposal"* **OR** *"concept"* **OR** *"idea"* **OR** *"method"* **OR** *"manner"* **OR** *"technique"* **OR** *"procedure" OR "program"* **OR** *"assistant")*

We use the search string in five scientific databases, adapting the search string to the specific database requirements—e.g., limiting the number of wildcards (*). After executing the database search strategy, we use the procedure proposed by Wohlin [19] for implementing the snowballing search strategy. First, we need to select an initial set of records for starting the snowballing. We use the inclusion/exclusion criteria (see Section 3.3) to select the first set of possible proposals from the database search. Then, we choose the top 10 records based on the quality assessment (see Section 3.4). Based on this first set of 10 records, we start iterating in rounds the references (backwards snowballing) and citing records (forward snowballing) using the inclusion and exclusion criteria. We stop iterating when no new records are found at the end of the round.

### 3.3 Inclusion/exclusion criteria

We propose the following set of inclusion/exclusion (I/E) criteria:

- **(I1)** Does the paper define a specific proposal for assisting users during modelling tasks in MDSE tools? Compilation of proposals like literature reviews, systematic mappings, or systematic literature reviews does not fulfil I1.

- **(I2)** Is the proposal designed to assist users during modelling tasks in MDSE tools? We focus on proposals that assist users during modelling tasks in MDSE tools, including—but not limited to—modelling, model tracing, model debugging, model repair, and model validation, among others. Proposals focused on assisting in developing MDSE tools do not fulfil I2.
- **(E1)** The proposal's main contribution is not on assisting users during modelling in MDSE tools. If assisting users during modelling tasks is not the main contribution, we exclude the proposal using E1—e.g., a proposal showing a new MDSE tool where assisting users during modelling tasks is superficially mentioned and is not the main goal of the proposal.
- **(E2)** The proposal is not related to software engineering.
- **(E3)** The proposal is not written in English.
- **(E4)** The proposal is not a peer-reviewed publication.
- **(E5)** The proposal's full text is not available.

Reviewers use I1-E5 to screen proposals' titles, abstracts, and full texts. Reviewers must agree that a proposal fulfils all inclusion criteria and does not fulfil any exclusion criteria to consider it as an included proposal.

### 3.4 Quality assessment strategy

Based on [20], we combine two types of quality assessments into a 3-Point-Liker-Scale questionnaire: subjective and objective quality assessments. For subjective quality assessment, reviewers answer questions based on their opinion about the proposal. For objective quality assessment, reviewers use the CORE ranking[2] for conferences, the Journal Citation Report[3] (JCR) for journals, and the proposal's number of citations. We show the quality assessment questionnaire in Table 1. Quality assessment provide us with subjective and objective data to ensure reliability and validity of the findings. Also, as we mentioned, we use the top 10 studies based on quality assessment results as the initial set of proposals for snowballing execution.

**Table 1.** 3-Point-Liker-Scale Questionnaire for Quality Assessment.

| **Subjective Questions** | 1 = Yes | 0 = Partially | -1 = No |
|---|---|---|---|
| 1. Is the proposal for assisting users during modelling tasks in MDSE tools clear? | | | |
| 2. Are the proposal's limitations clear? | | | |
| 3. Are the proposal's goals clear? | | | |
| 4. Are the proposal's tools/sources downloadable? | | | |
| 5. Is there a clear case study or example illustrating the proposal? | | | |
| 6. Is the whole proposal empirically evaluated? | | | |
| 7. Are the users clearly described? | | | |
| 8. Are the results clearly explained? | | | |
| **Objective Questions** | | | |
| 9. How important is the proposal's publication venue? | | | |
| | 1 = Very important | 0 = Important | -1 = Not that important |
| CORE Range: | A* - A - B | C | Not indexed |

[2] http://portal.core.edu.au/conf-ranks/

[3] https://www.scimagojr.com

| | | | |
|---|---|---|---|
| JCR Range: | Q1 - Q2 | Q3 - Q4 | Not indexed |
| 10. How many citations do the proposal have? | | | |
| | 1 = More than 4 | 0 = Between 2 and 4 | -1 = Less than 2 |

### 3.5 Data extraction strategy

We aim to extract data from our systematic mapping studies based on RQ1, RQ2, and RQ3. Reviewers must extract the text fragment(s) that contains possible answers for each RQ as follows:

- RQ1: How is software modelling assisted? Extract the keywords the proposals' authors use for describing the proposal's strategy to assist during modelling tasks—e.g., methodology, technique, framework, among others.
- RQ2: What goals and limitations do existing MDSE tools report in terms of modelling assistance? Extract all goals and limitations the authors state proposal has. If authors do not state any goal or limitation, leave the field blank.
- RQ3: Which evaluation metrics and target users do existing MDSE tools consider for modelling assistance? Extract if the authors empirically evaluate the proposals, which evaluation metrics they use for the evaluation, and which target users they expect to use their proposal. Leave the field blank if the authors do not state something about this information, or if the authors use "user" to refer to their target users.

After extracting the literal data from the studies comprising RQ1, RQ2, and RQ3, we proceed to cluster and analyse them. We utilized the terminology used by the authors to create these clusters and carried out a cluster-level analysis of the data. It is important to note that this approach may introduce bias in data extraction and subjective interpretation. We extensively discuss these validity threats to our study in Section 6.

## 4 Systematic mapping study results[4]

### 4.1 Systematic mapping study protocol execution results

Three reviewers were involved in the proposal selection—a.k.a. primary study selection. The first reviewer (R1) has more than four years of experience in model-driven-development-related scientific topics (first author of this paper). The other two reviewers (R2 and R3) are Computer Science MSc students with limited knowledge of model-driven development. However, we provide them with the necessary tools and training for conducting the proposal selection. Moreover, we introduce a fourth reviewer (R4)

[4] The data extracted from proposals is mainly text, sometimes full paragraphs. Therefore, we have pre-processed these data to allow us to summarize the results and present them in the paper. Thus, the data presented in this paper are the preprocessed data for discussion and analysis. However, the clustering was performed using the complete data. Furthermore, for the sake of transparency, replicability, and confidence in our study, we have published the raw data in a public repository that can be consulted here: https://zenodo.org/records/10262145.

that has more than 10 years of experience in model-driven-development-related scientific topics (second author of this paper) to review the resulting clusters to mitigate the subjective interpretation validity threat (see Section 6.1).

First, we executed the database search using the search string (see Section 3.2), resulting in an initial set of 1.806 records. Then, R1 and R2 reviewed the first 1.806 records using the inclusion/exclusion criteria (see Section 3.3), resulting in 51 possible proposals. R1 and R2 assessed the quality (see Section 3.4) of these 51 possible proposals and selected the top 10 for executing the snowballing search strategy (see Section 3.2). After four rounds of backward and forward snowballing, R1 and R2 reviewed 807 records more. In total, R1 and R2 reviewed 2.613 records resulting from the database and snowballing search strategies. As a result, R1 and R2 selected 63 possible proposals. Then, R3 exhaustively reviewed the set of 63 possible proposals to validate the selection that R1 and R2 did. R3 and R1 discussed the inclusion/exclusion of the 63 possible proposals, resulting in a final set of 44 proposals [21–65]. To measure the reliability of agreement between reviewers, we measured the inclusion reliability using the K-Statistic [14, 15] between the first and final proposal selection. As a result, we observe a K-statistic = 0.614, showing a substantial agreement based on Landis and Koch's interpretation of the K-statistic [15] (0.80 > K-statistic > 0.61). Then, R1 proceeded to cluster the extracted data for the accepted proposals. After the clustering, R4 reviewed the extracted data and proposed clusters reporting disagreements. We measure again the K-statistic between the first clustering and the resulting clustering after R4 review. We observe a K-statistic = 0.709, showing a substantial agreement based on Landis and Koch's interpretation of the K-statistic [15] (0.80 > K-statistic > 0.61) for the clustering result. Including several reviewers and the clustering raise some internal validity threats. We discuss such threats in Section 6.1. We summarise the proposal selection and the quality assessment results in Fig. 3. In the following subsections, we provide a deeply description about resulting clusters for each RQ.

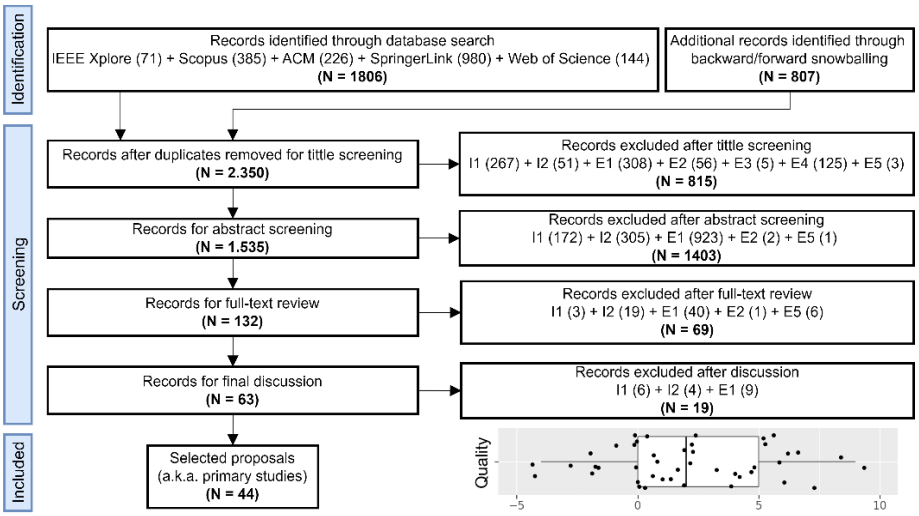



**Fig. 3.** PRISMA Flow chart for proposal selection and quality assessment results.

### 4.2 RQ1: How is software modelling assisted?

After analysing the extracted data for RQ1, we propose six clusters emerged from the study and based on keywords from each proposal: *tools*, *techniques*, *methods*, *frameworks*, *guidelines*, *and languages.* We show each cluster with the corresponding keywords in Table 2. Moreover, we show the distribution of strategies for assisting users during modelling tasks in MDSE tools in Fig. 4.

**Table 2.** RQ1: How is software modelling assisted? Identified clusters in our study.

| **RQ1: How is software modelling assisted?** | |
|---|---|
| **Clusters** | **Keywords** |
| *Tools* | Domain modelling recommender system [39], artificial intelligence-empowered software assistants [33], view manager with a meta layout [61], generic modelling environments [26, 65], adaptable modelling environments [31], reactive systems [35], model testing tools [49, 59], transformation-based tools [41, 62], and collaborative management tools [30]. |
| *Techniques* | Model development techniques [51, 55], model validation techniques [23, 43, 45, 46, 53], and model repair techniques [36, 57]. |
| *Methods* | Consistency validation methods [25, 29], service-oriented methods [22], model repair methods [21], task-driven methods [37, 40], and model-driven engineering methods [58, 64]. |
| *Frameworks* | Agent-based change propagation frameworks [24, 63], collaborative modelling frameworks [48], co-evolution frameworks [44], formal frameworks [60], and modelling frameworks [28, 54, 56]. |
| *Guidelines* | ISO-based standardisations [32], flexible workflows [42], refactoring processes [38], and multi-modelling architectures [27]. |
| *Languages* | Mega-modelling languages [50], language extensions [47], and modelling templates [34]. |

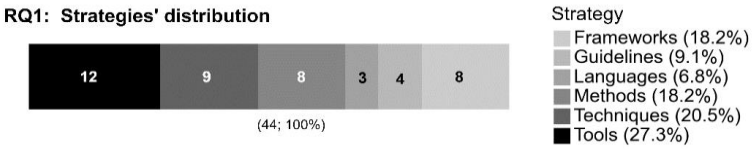



**Fig. 4.** RQ1: Strategies for modelling assistance distribution.

We create each cluster using the keywords the authors refer to their proposal after reviewing full text. That is the case for *techniques*, *methods*, and *frameworks*, where authors labelled their proposals with such keywords. On the other hand, there where proposals that do not use a keyword for identifying their proposal—e.g., defining their proposals as an "approach." Therefore, we required to include extra criteria for considering a proposal part of such clusters in the case of *tools*, *guidelines*, and *languages*. For *tools*, we cluster proposal where authors refer to their proposals as a *tool* or any keyword that refers to a software implementation—e.g., system, software assistant, environment, manager. For *guidelines*, we cluster proposals that establishes where, when,

how perform the modelling assistance based on standards, workflows, processes, or architectures. For *languages*, we cluster proposals where the authors propose modifying or creating a modelling language for allowing modelling assistance—e.g., language definition, minimal extensions, template application. We recognize that in software engineering research, definitions of *method*, *framework*, *technique*, and *tool* are still not unified—i.e., an author can consider their proposal as a tool, other authors can consider the same proposal as a technique. In our study, we rely on the keywords adopted by proposals' authors. However, we acknowledge this lack of common vocabulary as a threat to validity and possible future challenges in this area. We mention this in Section 6.

We observe *tools* are the most common strategy to assist during modelling (27.3%); followed by *techniques* (20.5%), *frameworks* (18.2%) and *methods* (18.2%). *Guidelines* (9.1%) and *languages* (6.8%) are the less common strategies to assist during modelling. Based on such distribution, we observed that 93.3% of proposals assist during modelling using totally or partially software implementations—e.g., recommender systems (*tool*), model validation techniques (*technique)*, model repair methods (*method)*, agent-based change propagation frameworks (*framework)*, and modelling templates *(language)*. In contrast, 9.1% of proposals assist users to model software using *guidelines (9.1%)*. *Guidelines* assist users during modelling, not necessarily using software, for instance, by standardising the steps users should follow to model. Therefore, we observe a trend in assisting users to model software using software-based strategies.

### 4.3 RQ2: What goals (G) and limitations (L) do existing MDSE tools report in terms of modelling assistance?

After analysing the extracted data for RQ2, we propose seven clusters about proposals' goals and five clusters about proposals' limitations. Moreover, we classify a proposal as L-NS when authors do not explicitly specify their limitations. We show each cluster with the corresponding keywords in Table 3. In Fig. 5 we show a bubble chart comprising limitations and goals clusters, also having their distribution among proposals.

**Table 3.** RQ2: What goals (G) and limitations (L) existing MDSE tools have? Identified clusters in our study.

| **RQ2: What goals (G) and limitations (L) existing MDSE tools have?** | |
|---|---|
| **Clusters** | **Keywords** |
| *G1. addressing change propagation* | Understand how model changes affect the model correctness [53], allow asynchronous changes [48], verify refactoring pre/post conditions [47], apply refactoring rules [38], detect inconsistent model changes [26, 36], support change-based process formalisations [30], achieve model change traceability [27, 28], and fix propagated changes [24, 63]. |
| *G2. enhancing consistency checking* | Keep models aligned [64], check between models and implementation [59], inspect inconsistencies in models [46, 57], ensure model intra/inter consistency [22, 25, 45, 58], and formalise consistency between models [29]. |
| *G3. ensuring model* | Ensure component-based software compatibility guarantees in open development and runtime environments [56]. |

| | |
|---|---|
| *compatibility* | |
| *G4. improving model quality* | Improve model quality by model validation, verification, and exploration [43]. |
| *G5. improving user interaction* | Enhance user interaction mechanisms [61], reduce users' cognitive load [55], alleviate editing modelling expressions [54], semantically lift low-level differences [65], organise models [50], make amenable formal verification [49], preserve user's creativity during modelling [42], guide users on modelling [35], reduce modelling complexity [32], allow end-users to cooperate from modelling stage [40], and make model management user friendly [23]. |
| *G6. easing model evolution* | Automatically generate models [31, 41, 51, 60], support model co-evolution [44], support domain modelling [39], support model reuse [37], decrease modelling time [34], and suggest model elements during modelling [21, 33]. |
| *G7. supporting vulnerability detection* | Detect model security vulnerabilities [62]. |
| *L1. accuracy* | Limitations on model automatic extraction accuracy [51] and inconsistency resolution accuracy [36]. |
| *L2. effort* | Proposals increase the effort on modelling maintenance [53], require modelling standardisation [32], and introduce additional manual modelling tasks [21]. |
| *L3. generality* | Limitations on specific domains [45, 62], specific refactoring types [47], and modelling scenarios [39, 46]. |
| L4. *learnability* | Users have problems learning uncommon modelling languages [64], learning how to make decisions [63], and learning underlying technologies [34]. |
| L5. *scope* | Lack specific modelling constraints [24, 57, 59], modelling components [56], and model types [29]. |
| *L-NS* | [22, 23, 25–28, 30, 31, 33, 35, 37, 38, 40–44, 48–50, 54, 55, 58, 60, 61, 65] |

**G:** goal; **L:** limitation; **NS:** Not specified; **MDE:** Model driven engineering

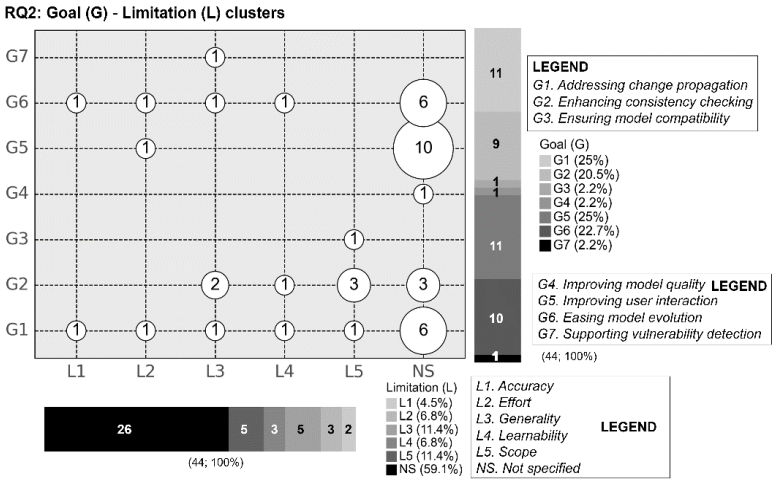



**Fig. 5.** RQ2: Goal (G) – Limitation (L) bubble chart and distributions.

*G1. Addressing change propagation.* We found proposals aiming to address change propagation during modelling tasks in MDSE tools. We frame change propagation as an umbrella comprising model synchronization, model change-based consistency checking, and change traceability during modelling tasks. For example, a proposal [53] aims to assist users during modelling tasks addressing change propagation by detecting how correct the models are when a change is introduced. Similarly, we cluster all proposals that aim to assist on easing, explaining, detecting, tracing, fixing, and verifying changes during modelling tasks in G1.

*G2. Enhancing consistency checking.* We found proposals aiming to enhance consistency checking during modelling tasks in MDSE tools. We observe that consistency checking during modelling tasks in MDSE tools involve not only inter-consistency—i.e., consistency between models inside MDSE tools—but also external consistency with other models, code, or requirements. For example, a proposal [57] aims to assist during modelling tasks in MDSE tools by detecting inconsistencies in models introduced by isolated editing of single views. Similarly, we cluster all proposals that aim to assist on checking, detecting, inspecting, showing, and ensuring consistency checking during modelling tasks in G2.

*G3. Ensuring model compatibility.* We found a proposal [56] aiming to ensure model compatibility during modelling tasks in MDSE tools. We observe that model compatibility is relevant when the resulting model will be used as a component-based software having other open development and runtime environments interacting with it. We cluster proposals that aim to assist on ensuring creating, generating, and producing compatible models in G3.

*G4. Improving model quality.* We found a proposal [43] aiming to improve model quality during modelling tasks in MDSE tools. We consider "model quality" as a broad term. In this proposal, we found model quality as a mix of verification, validation, and exploration. Thus, we cluster proposals that aim to assist on more than one quality aspect during modelling tasks such as validation, verification, and exploration in G4.

*G5. Improving user interaction.* We found proposals aiming to improve user interaction during modelling tasks in MDSE tools. We observe that user interaction with MDSE tools involve how user effectively use MDSE tools to achieve their goals. For example, a proposal [65] aims to assist users during modelling tasks improving user interaction by providing user-comprehensible specification of changes between two models. Similarly, we cluster all proposals that aim to assist on alleviating, easing, supporting, and enhancing user interaction during modelling tasks in MDSE tools in G5.

*G6. Easing model evolution.* We found proposals aiming to ease model evolution in MDSE tools. We observe that model evolution involves implementing model instances, creating models, co-evolving models based on historical data, or editing existing models. For instance, a proposal [21] aims to assisting users during modelling tasks easing model evolution by generating repair plains when an error/bug is triggered. Similarly, we cluster all proposals that aim to assist on supporting, implementing, and generating model evolutions in G6.

*G7. Support vulnerability detection.* We found a proposal [62] aiming to support vulnerability detection during modelling tasks in MDSE tools. We observe that vulnerability detection is relevant for ensuring that the resulting software complies with no-

functional requirements related to security, performance, and efficiency, among other quality attributes. This proposal's goal is to detect security vulnerabilities and suggest corrective actions to address these weak points during modelling tasks. Thus, we cluster proposals that aim to assist on identifying, correcting, and detecting vulnerabilities during modelling tasks in MDSE tools in G7.

*L1. Accuracy*. We found proposals reporting accuracy limitations. We cluster proposals in L1 when the authors state the obtained results is not the expected result. For instance, a proposal [51] reports accuracy limitations on the results of automatic model extraction. We cluster all proposals that report a lack of accuracy in L1.

*L2. Effort.* We found proposals reporting effort limitations. We cluster proposals in L2 when the authors state their proposal requires increasing effort before, during, and after modelling tasks. For example, a proposal [32] reports effort limitations since their proposal require previously standardizing all the MDSE tool architecture metamodel. We cluster all proposals that report similar effort limitations in L2.

L3. *Generality.* We found proposals reporting generality limitations. We cluster proposals in L3 when the authors acknowledge that—even if their work contributes a novelty and advance on the state of the art—their resulting proposal is limited to specific modelling scenarios—i.e., their results/proposal cannot be generalized. For instance, some proposals [45, 62] report generality limitations due to their proposal is limited to a specific domain. We cluster all proposals that report similar generality limitations in L3.

*L4. Learnability.* We found proposals reporting learnability limitations. We cluster proposals in L4 when the authors state their proposal require users to learn extra skills. For example, a proposal [64] reports learnability limitations due to users need to learn a new no-well known modelling language in a business environment. We cluster all proposals that report similar learnability limitations in L4.

*L5. Scope.* We found proposals reporting scope limitations. We cluster proposals in L5 when the authors acknowledge that their proposal lack specific features. For instance, some proposals [24, 57, 59] report scope limitations due to their proposal lack support for specific consistency constraints during modelling tasks. We cluster all proposals that report similar scope limitations in L5.

Based on what we observe, existing proposals mostly assist users in refining existing models in MDSE tools. In total, 22.7% (10) proposals aim to assist users specifically in creating models by *easing model evolution (G6).* On the other hand, most proposals aim (23, 52.3%) to refine already existing models by *addressing change propagation (G1), enhancing consistency checking (G2), ensuring model compatibility (G3), improving model quality (G4), and supporting vulnerability detection (G7).* Finally, there are proposals (11, 25%) that aim to assist users in both creating and refining models by *improving user interaction (G5)* in MDSE tools.

Moreover, we observe that most proposals do not specify their limitations, which hinders research gap identification. Less than half of proposals (18, 40.9%) explicitly specify *limitations.* This also raises inconsistencies. For instance, most proposals aiming to *improve user interaction (G5)* do not specify any limitations. Even though *improving user interaction* is still an unaddressed challenge based on the literature, users,

and MDSE experts [66]. Regarding the relationship between goal clusters and limitation clusters, we observe that *G2 (enhancing consistency checking)* shows a relevant relationship, where more than a half (5; 55.6%) of the proposals report *generality (L3)* and *scope (L5)* limitations. On the other hand, we observe that other proposals are equally distributed over the limitations, such as those classified in *G1 (addressing change propagation)* and *G6 (easing model evolution)*. *G7 (supporting vulnerability detection)* and *G3 (improving model quality)*. However, they contain just one proposal each, limiting the analysis. Finally, as we mentioned before, *G5 (improving user interaction)* and *G4 (improving model quality)* have most proposals without explicitly specifying limitations.

### 4.4 RQ3: which evaluation metrics (M) and target users (U) do existing MDSE tools report for modelling assistance?

After analysing the extracted data for RQ3, we propose three clusters about proposals' metrics and four clusters about proposals' target users. In the case of metrics, we classify the metrics we found using the Technology Acceptance Model (TAM) [16]. Notice that a proposal could have more than one evaluation metric specified. For users, we cluster them based on the extracted text. Moreover, we cluster proposals that do not specify an evaluation metric when they were not evaluated (NE). On the other hand, we cluster proposals as user not specified (U-NS) when authors use the generic keyword "user" or refer to "he/she" as the intended user to assist with their proposal. We show each cluster with the corresponding keywords in Table 4. In Fig. 6 we show a bubble chart comprising evaluation metrics and target users, also having their distribution among proposals.

**Table 4.** RQ3: Which evaluation metrics (M) and target users (U) existing MDSE tools consider? Identified clusters in our study.

| **RQ3: Which evaluation metrics (M) and target users (U) existing MDSE tools consider?** | |
|---|---|
| **Clusters** | **Keywords** |
| *M1. effectiveness* | Number of detected real faults [59], the degree of stakeholder participation [64], feasibility of execution [58], inconsistency coverage [57], success scores [55], accuracy [21, 41, 53], compression factor [65], effectiveness [63], and precision [25]. |
| M2. e*fficiency* | Modelling time [55, 61], model completion time [60], testing generation time [59], model repair generation time [57], performance [54, 62, 63, 65], computational effort [53], and execution time [21]. |
| M3. *user perception* | Perception of industrial adoption [58] and perceived usefulness [30]. |
| *NE* | [22–24, 26–29, 31–40, 42–51, 56] |
| U1. d*esigners/modellers* | Software designers [63], model developers [60], professional engineers with design experience [58], designers [32], modellers [39, 45], and MDE developers [49]. |
| U2. *domain experts* | Business analysts [64], end-users [26, 40], domain users [61], domain experts [42], and business users [30]. |

| | |
|---|---|
| U3. *software developers* | Developers [25, 27, 31, 34, 36, 41, 43, 44, 47, 48, 50, 51, 57], software developers [56], UML developers [21], software engineering students [55] and software maintainers [24]. |
| *U-NS* | [22, 23, 28, 29, 33, 35, 37, 38, 46, 53, 54, 59, 62, 65] |

**M:** metric; **U:** Target user; **NS:** Not specified; **NE:** Not evaluated; **MDE:** Model driven engineering

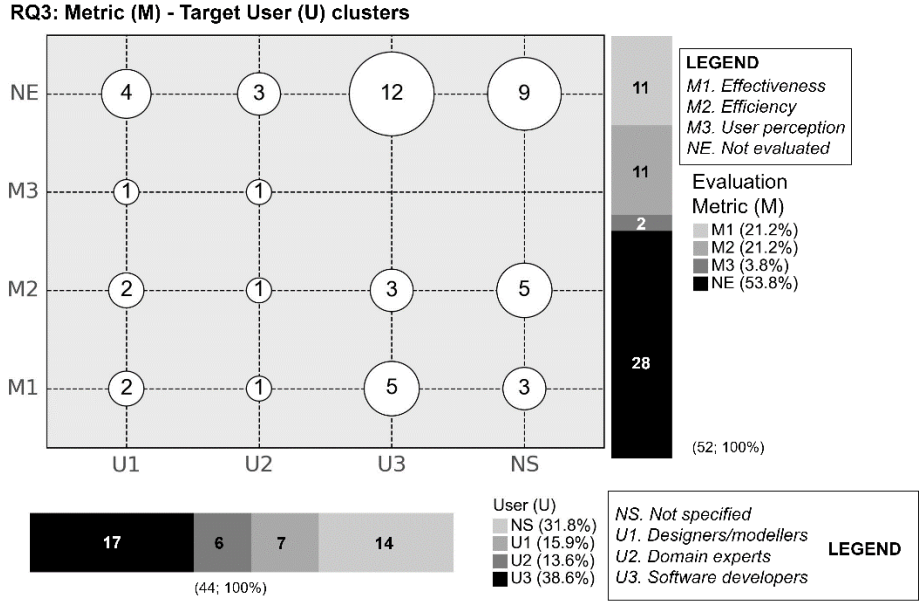



**Fig. 6.** RQ3: Metric (M) – Target User (U) bubble chart and distributions.

*M1. Effectiveness.* Based on TAM [16], effectiveness metrics intent to measure the degree to which a proposal achieves its objectives. Thus, we cluster all proposals that explicitly specify metrics that fits with that definition in M1. For example, a proposal [59] that aim to assist on checking where there is an inconsistency between models and code propose using "number of detected real faults" as evaluation metric. We cluster this metric into M1since the metric reflects how effective is the proposal on detecting inconsistencies and faults between model and code—i.e., the proposed metric shows the degree on which the proposal meets its objectives.

*M2. Efficiency.* Based on TAM [16], efficiency metrics intend to measure the effort to apply a proposal. Thus, we cluster all proposals that explicitly specify metrics that fits with that definition in M2. For instance, some proposals [55, 61] use as evaluation metric the "modelling time." We cluster this metric into the M2 since "modelling time" is a metric that represents how much effort user needs to complete a modelling task.

*M3. User perception.* Based on TAM [16], user perceptions are divided into perceived ease of use, usefulness, and intention to use. Then, user perception metrics can measure the degree a person believes that using a particular proposal would be free of

effort; the degree a person believes that a particular proposal will be effective in achieving its intended objectives; or to what extent a person intends to use a particular proposal. Thus, we cluster all proposals that explicitly specify metrics that fits with one or more of such definitions in M3. For example, a proposal [58] uses "perception of industrial adoption" as evaluation metric. We cluster this metric into M3 since "perception of industrial adoption" measure to what extend a person in industry would adopt the proposal.

*U1. Designers/modellers.* We found proposals that consider *designers or modellers* as their target users. We cluster proposals into U1 when their target users' keywords refer to a person who specify models/designs in MDSE tools. For example, a proposal [60] states that their target users are "model developers." We cluster this proposal into U1 since model developers' main goal developing models in an MDSE tool.

*U2. Domain experts.* We found proposals that consider *domain experts* as their target users. We cluster proposals into U2 when their target users' keywords refer to a person who has domain expertise in the modelled business but not necessarily in modelling. For instance, a proposal [30] states that their target users are "business users." We cluster this proposal into U2 since business users have expertise in their business domain—e.g., healthcare, finances, insurance—but not necessarily in modelling.

*U3. Software developers.* We found proposals that consider *software developers* as their target users. We cluster proposals into U3 when their target users' keywords explicitly contain the keyword *developer* and/or involves a role relating to software engineering that do not fit in U1 or U2. For example, a proposal [21] states that their target users are "UML developers." We cluster this proposal into U3 since their target user contains explicitly the word "developer." Notice that "UML developer" could be consider also as part of U1 since UML is a modelling language. However, we stick to the hard rule of the "developer" keyword for classifying in U3 since it is not clear whether the goal of this role would be developing a UML model or developing software using their knowledge of UML.

Based on what we observe, most proposals have been evaluated using objective metrics, leaving aside subjective user perceptions. The most common evaluation metrics (22; 42.4%) around proposals are effectiveness (M1; 11; 21.2%) and efficiency (M2; 11, 21.2%). This shows a trend toward measuring objective evaluation metrics—i.e., metrics that do not require direct input from the user. On the other hand, user perception metrics (M3) are a minority (2; 3.8%). However, most of the proposals define no metric/were not evaluated (28; 53.8%). In terms of target users, we observe *software developers* (U3) are the most common target users among proposals (17; 38.6%), followed by *designers/modellers* (U1; 7; 15.9%). In contrast, we observe *domain experts* (U2; 6; 13.6%) are the less common target users among proposals. Finally, we notice that there are proposals (14; 31.8%) that do not specify any target user.

# 5 State of the practice on modelling assistance in MDSE tools[5]

Up to this point we have explored studies in the literature on modelling assistance based on a systematic mapping study. This view is relevant, as it is research that guides innovation in the industry. On the other hand, MDSE is a widely practical field. This implies that the literature alone does not contains all current advancements in modelling assistance. Therefore, we consider relevant to review the state of the practice on how MDSE tools have proposed to assist their users during modelling tasks. Thus, we introduce the following research question (RQ):

- **RQ4.** *What is the state of the practice on modelling assistance?* To answer this question, we aim to gather data from MDSE tools used in practice. Thus, we expect to extract descriptions of proposals for assisting during modelling tasks in such MDSE tools.

First we plan to explore which MDSE tools are available in practice to answer RQ4. To do so, we rely on Gartner Magic Quadrant (GMQ) for enterprise low-code application platforms [17]. GMQ is a culmination of research in a specific market, giving a wide-angle view of the relative positions of the market's competitors [67]. We explore all MDSE tools mentioned in the GMQ looking for their documentation. Then, we review their documentation, extracting quotes about how they propose to assist users during modelling tasks. We show graphically these steps, including inputs and outputs, in Fig. 7.

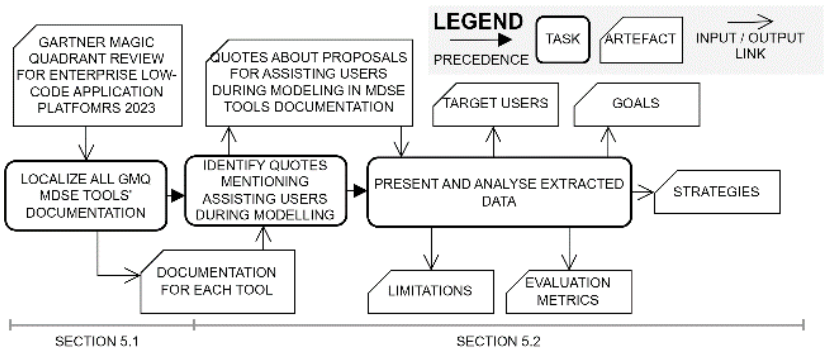



**Fig. 7.** Overview on Reviewing MDSE tools.

[5] As the systematic mapping results, the data extracted from MDSE tools in this section from documentation, websites, and user guides is mainly text, sometimes full paragraphs. Therefore, we have pre-processed these data to allow us to summarize the results and present them in the paper. Thus, the data presented in this paper are the preprocessed data for discussion and analysis. For the sake of transparency, replicability, and confidence in our study, we have published the raw data in a public repository that can be consulted here: https://zenodo.org/records/10262145.

### 5.1 MDSE tools' review: Gartner Magic Quadrant for enterprise low-code application platforms

Based on the GMQ for enterprise low-code application platforms in 2023 (see Fig. 8), there were 17 tools classified into challengers (1), leaders (5), niche players (8), and visionaries (3). In the following paragraphs, we introduce each cluster and its meaning based on GMQ, followed by a list of the MDSE tools classified in such cluster. In our study, we use GMQ as a research-supported and industry-relevant market study to gather grey literature on MDSE tools documentation. However, we acknowledge that there is other grey literature that could have been included but were not, such as reports and white papers. We mention this as a validity threat in Section 6.

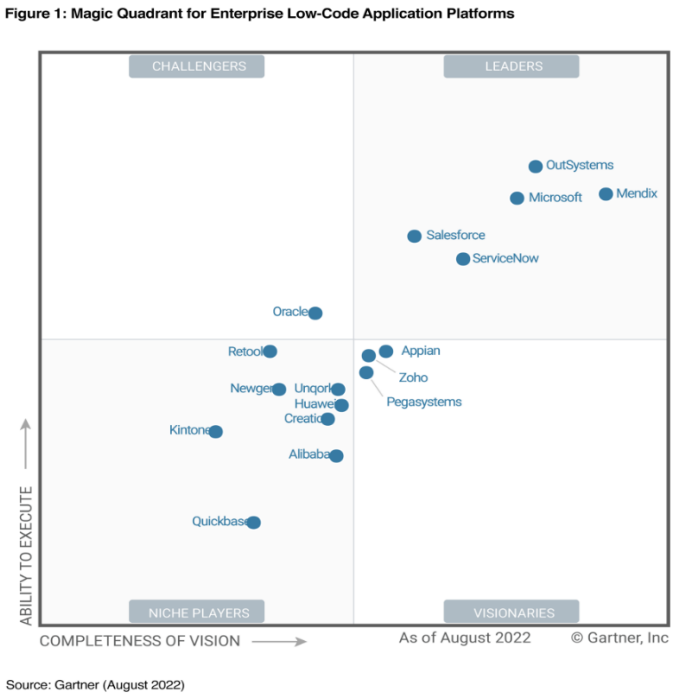



**Fig. 8.** Gartner Magic Quadrant for enterprise low-code application platforms. Source: [17]

*Leaders (LE).* MDSE tools classified as *leaders* execute comparatively well today and they are well positioned for tomorrow in the market [67]. We find five MDSE tools in such classification: OutSystems [68], Mendix [69], Microsoft Power Apps [70], Salesforce [71], and ServiceNow [72].

*Challengers (C).* MDSE tools classified as *challengers* executes comparatively well today or may dominate a large segment but does not have a roadmap aligned to Gartner's view of how a market will evolve [67]. We find just Oracle APEX [73] classified as *challenger*.

*Visionaries (V).* MDSE tools classified as *visionaries* understand where the market is going or has a vision for changing market rules but do not yet execute comparatively well or do so inconsistently [67]. We find three MDSE tools classified as *visionaries:* Appian [74], Zoho Creator [75], and PegaSystems [76].

*Niche players (NP).* MDSE tools classified as *niche players* focus comparatively successfully on a small segment or are unfocused and does not out-innovate or outperform others [67]. We find eight MDSE tools classified as *niche players:* Retool [77], NewGenONE [78], Unqork [79], Huawei Astro Zero [80], Creatio ONE [81], YiDA by Alibaba [82], Kintone [83], and Quickbase [84].

### 5.2 RQ4: What is the state of the practice on modelling assistance?

We explore documentation, websites, and user guides from each MDSE tool mentioned in each GMQ 2023 cluster—i.e., leaders (LE), visionaries (V), challengers (C), and niche players (NP)—to answer RQ4. We search for insights into this data about how they aim to assist users during modelling tasks in such tools. Then, we extract the text and classify them following the same relevant data we extracted from RQ1, RQ2, and RQ3—i.e., we classify the text if we consider that the authors are referring to a strategy (S), goal (G), limitation (LI), evaluation metric (M), or target user (U). In Table 5, we show the quotes from MDSE tools' documentation for each GMQ MDSE tool and how we classify them.

**Table 5.** RQ4: What is the state of the practice on modelling assistance? Strategies (S), goals (G), limitations (LI), evaluation metrics (M), and target users (U) based on quotes from GMQ MDSE tools' documentation

| MDSE Tool | Modelling assistance keyword | Quotes from documentation (i.e., websites, user guides, and technical documentation) |
|---|---|---|
| Mendix (LE) | MXAssist Logic Bot | "…Help development teams model and deliver Mendix applications faster…" (G) |
| | | "…Recommends the top five next best actions out of more than 40 different options…" (S) |
| | | "…continuous improvement and training of the model has elevated the accuracy level to 95%..." (M) |
| | | "…when the developer inserts a new element…" (U) |
| | MxAssist Performance Bot | "…assists you in improving the performance of your app by inspecting your app project model against Mendix development best practices…" (G) |
| | | "…the bot inspects the model, identifies issue, and pinpoints the document/element causing the issue…" (S) |
| | | "…the bot explains the identified issue, potential impact, and shows the way to fix it…" (S) |
| | | "…MxAssist Performance Bot enhances development efficiency by substantially reducing peer reviews…" (M) |
| | | "…increases developer productivity via the automatic detection and pinpointing of issues…" (M) |
| | | "…educates junior developers on best practices…" (U) |

| | | |
|---|---|---|
| | Validation Assist | "…helps you build validation microflows in a more automated way using pre-built expressions…" (G) |
| | | "…List of checks for all members which data type can be empty…" (S) |
| | | "…Auto-generation of the validation microflow…" (S) |
| OutSystems (LE) | AI Code Mentor | "…assistant that guides you through the OutSystems platform, dramatically accelerating and improving application development..." (G) |
| | | "…this AI-powered capability is especially handy when you know which data you want to get from the database but are unsure how to add it to an aggregate…" (S) |
| | | "…The feature may not work when you have a big number of entities and attributes…" (L) |
| | | "…You can only ask for data using English…" (L) |
| | | "…Typos in the name of entities, attributes, and variables may cause errors in the generated aggregate…" (L) |
| | | "…You can't ask for data filtered with advanced Functions…" (L) |
| | | "…You can't ask for data ordered using dynamic sorts…" (L) |
| | Application templates | "…Thanks to the application templates, the apps have many predefined elements that save you development time…" (G) |
| | | "…Forge application templates…" (S) |
| | | "…save you development time…" (M) |
| | Architecture Mentor | "…AI Architecture Mentor makes sure your code meets critical architectural standards…" (G) |
| | AI Security Mentor | "…AI-based security lead on your team to constantly review code to check for vulnerabilities…" (G) |
| | AI Performance Mentor | "…Wishing you had a performance expert on your team to constantly review code, ensuring that apps will consistently perform and scale at peak levels? Now you do…" (G) |
| | AI Maintainability Mentor | "…reduce technical debt and ensure best practices are followed…" (G) |
| | | "…scans your entire app portfolio…" (S) |
| Microsoft Power Apps (LE) | Copilot | "…Create Power Apps with the help of AI…" (G) |
| | | "…You can tell the AI assistant what kind of information you want to collect, track, or show and the assistant will generate a Dataverse table and use it to build your canvas app…" (S) |
| | | "…Preview features aren't meant for production use and may have restricted functionality..." (L) |
| Salesforce (LE) | Performance assistant | "…Use the step-by-step instructions, articles, and tools to help you architect your system, conduct performance testing, and interpret your results…" (G) |
| | | "…central hub of information and resources about scalability and performance testing with Salesforce…" (S) |
| | Einstein GPT for Salesforce Developers | "…automating the creation of repetitive code elements helps ensure consistency and standardization in the codebase…" (G) |
| | | "…this not only speeds up the development process, but also reduces the risk of human error…" (M) |
| | | "…generative code generation can speed up the prototyping process by quickly creating boilerplate code…" (S) |
| | | "…it can save developers a lot of time…" (U) |

| | | |
|---|---|---|
| Appian (V) | Appian Developer Co-Pilot | "…helps developers while building process models by suggesting the nodes that we predict they are most likely to use next…" (G) |
| | | "…To help you, Appian has added "Suggested" nodes…" (S) |
| | | "…That AI then helps developers…" (U) |
| Oracle APPEX (C) | Intelligent Wizard | "…guide you through the rapid creation of applications and components…" (G) |
| | | "…Intelligent wizards to guide…" (S) |
| Retool (NP) | AI Features | "…help you write and edit SQL queries using natural language…" (G) |
| | | "…Retool's AI operates in three modes: generate, edit, and explain…" (S) |
| | | "…Retool cannot guarantee that the output is always accurate…" (L) |
| **Not assistant found** | ServiceNow (LE), Zoho Creator (V), PegaSystems (V), Newgen (NP), Unqork (NP), Huawei (NP), Creatio (NP), YiDA by Alibaba (NP), Kintone (NP), Quickbase (NP) | |

**LE**: Leaders; **V**: Visionaries; **C**: Challengers; **NP**: Niche Players; **S**: Strategy; **G**: Goal; **L**: Limitation; **M**: Evaluation metric; **U**: Target user

After reviewing seventeen MDSE tools we found in the GMQ we observe that most of them do not mention how they intend to assist users during modelling in their documentation (10; 58.8%; see Fig. 9A). In addition, of the seven tools (58.7%) we observed that mention how they assist their users during modelling tasks in the documentation, most of them are part of the market leaders (LE; 4; 23.5%; see Fig. 9B) according to the GMQ. On the other hand, at least one MDSE tool in each GMQ mentioned how they intent to assist users during modelling tasks in their documentation (see Fig. 9B).

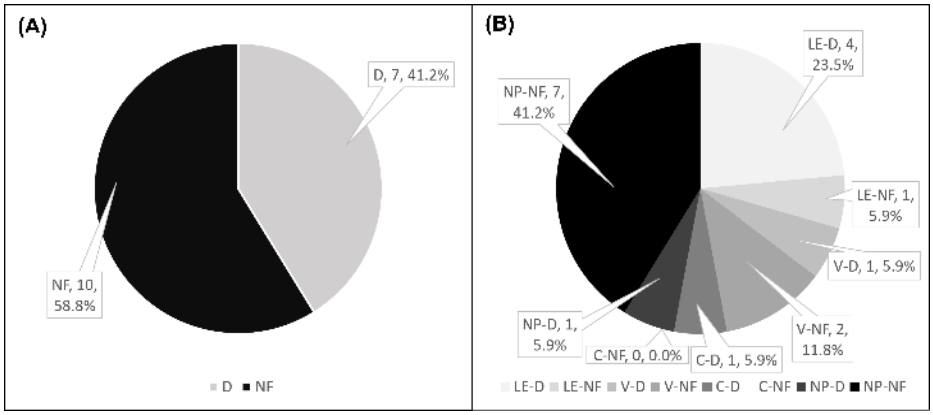



**Fig. 9.** (**A**) Distribution of GMQ MDSE tools with documentation on assisting during modelling (D) and not documentation found (NF). (**B**) Distribution of GMQ MDSE tools with documentation on assisting during modelling (D) and not documentation found (NF) per GMQ classification: Leaders (LE); Visionaries (V); Challengers (C); and Niche Players (NP).

Furthermore, we found fifteen proposals for assisting users during modelling inside the seven MDSE tools' documentation. In Fig. 10 we show the distribution of proposals mentioning strategies, goals, limitations, evaluation metrics, and target users.

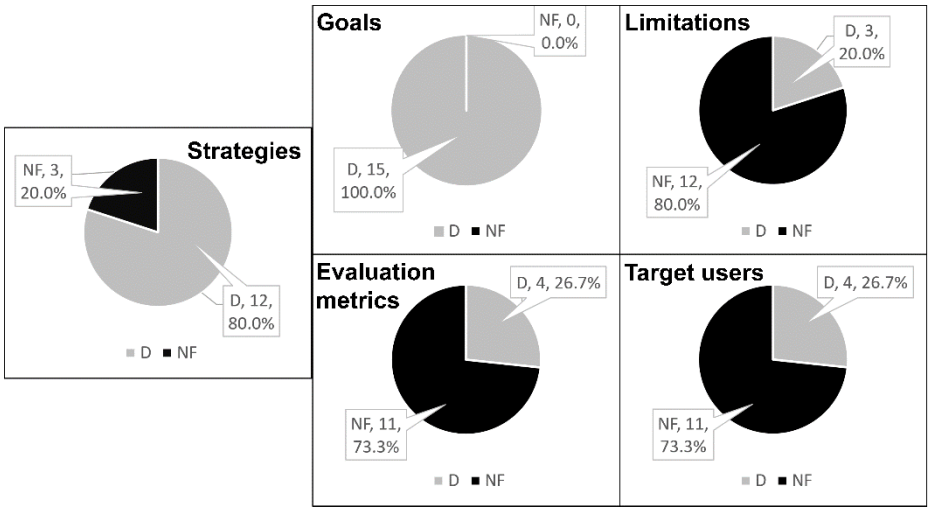



**Fig. 10.** Distributions on MDSE proposals for assisting users during modelling that mention (D) strategies, goals, limitations, evaluation metrics, and target users and which ones that not (NF).

We observe that most proposals (12; 80.0%) mention the strategy they follow for assisting during modelling, including keywords such as: AI-power capabilities, templates, scanners, recommenders, model inspectors, AI assistants, and intelligent wizards. In terms of goals, we observe all proposals (15; 100.0%) mention at least one goal they want to achieve by assisting users during modelling tasks, including keywords such as: improving application development, accelerating application development; ensure code meets critical architectural standards; check for vulnerabilities; ensuring performance; validate models; create models; and speed up prototyping process. In contrast, most proposals do not specify limitations (12; 80.0%). The proposals that state limitations are a minority (3; 20.0%), mentioning limitations such as: preview features are not meant for production, or the tool cannot guarantee that the output is always accurate. Regarding evaluation metrics, most proposals have not specified any metric (11; 73.3%). Specified evaluation metrics among proposals include development time, accuracy, efficiency, productivity, and risk of human error. Similarly, most proposals have not defined a target user (11; 73.3%). Specified target users include developers and junior developers. It is important to mention that due to the way the authors write their documentation, it is difficult to identify a specific target user. We found that authors write MDSE tools' documentation using second person, instead of third person—i.e., authors write using the subject "you." This hides the actor performing the action—i.e., hides the target user.

# 6 Threats to validity and limitations

This section presents a discussion about the internal validity, construct validity, and external validity threats of the presented results. We highlight the limitations and how we tried to mitigate them. In addition, we discuss and point out the limitations that we have not been able to mitigate and should be considered when interpreting the results of our study.

## 6.1 Internal validity

Internal validity is defined as the extent to which the study design, methodology, and data collection process accurately capture and represent the identified studies. We have identified three internal validity threats on our study: *selection bias, data extraction bias, and inter-rater reliability.*

We acknowledge that selecting relevant studies and MDSE tools for inclusion our study may introduce a *selection bias*. To mitigate this threat, we have defined a set of exclusion and inclusion criteria, so the reviewers' selection is as objective as possible. Moreover, we rely on a research-supported and industry-relevant market study-—i.e., GMQ—to select the MDSE tools that we use for answering RQ4. However, the bias of the GMQ and reviewers' subjective judgement to select the MDSE tools could result in other relevant studies not being selected.

Data extraction during our study introduces a risk of bias in how the data are collected, introducing a *data extraction bias validity threat*. To mitigate this threat, we ask reviewers to extract as much text as possible that could have relevant information around the specific RQ they were extracting data for. Regarding the MDSE tool data extraction, we rely on public available documentation, and we extracted as much data as possible from such documentation. Moreover, the first and second authors of this paper reviewed the extracted data and supervise that the data other reviewers extracted seems to be as complete as possible. Nevertheless, human error is inevitable. In addition, we recognize that especially MDSE tools have limited documentation on proposals for assisting during modelling tasks. This means that such MDSE tools may implement modelling assistance proposals but not necessarily document them. This causes our extracted data on MDSE tools in the market to be limited to documented proposals to assist users during modelling tasks. Finally, we consider relevant to mention that our study relies on the authors' terminology from both the literature and the MDSE tool documentation. This means that the data collected could be biased to the authors' terminology leading to inconsistencies. For example, some authors define their proposal as a tool, but other authors may consider the same proposal as a technique. This shows the need to establish a common terminology in our field to avoid these terminological problems.

We identified a validity threat of subjective interpretation regarding the clusters we proposed in our study. As mentioned earlier, we primarily relied on terminology extracted directly from authors' studies to propose clusters that represent, as closely as possible, the studies grouped within them. However, we noticed that the ontological background for proposing such clusters appears to be subjective and open to various

interpretations, potentially introducing bias into our research findings. To address this concern, we implemented rigorous data collection and analysis methods. Moreover, we conducted a triangulation after proposing each cluster. We have measured the level of agreement between the first clustering and after the triangulation. We observe a K-statistic = 0.709, showing a substantial agreement between the first clusters and final cluster discussion based on Landis and Koch's interpretation of the K-statistic [15] (0.80 > K-statistic > 0.61). Nevertheless, it is crucial to recognize that the subjective nature of the ontological background may have influenced the interpretation of results. To mitigate this validity threat, we have reported our clustering process as transparently as possible, making available both data extraction and triangulation information. Moreover, this concern has prompted future work on validating the data we extracted, involving external reviewers and peer review processes.

Involving several reviewers to select studies introduces an i*nter-rater reliability threat.* We have involved three reviewers in the process and used the K-statistic to measure the effect of the *inter-rater reliability threat* in our results. We observed a K-statistic = 0.614, showing a substantial agreement between reviewers based on Landis and Koch's interpretation of the K-statistic [15] (0.80 > K-statistic > 0.61). Having calculated the K-statistic allow us to see that there is an agreement between reviewers for selecting studies. However, we did not measure K-statistic during data extraction since our data were mainly text. Despite this, we consider it important to emphasize that we have made several rounds with the authors of this paper to cluster the extracted data and reach an agreement between us as mentioned before. In addition, we recognize that this level of agreement may be affected by reviewer fatigue. To this end, the reviewers had enough time to extract the data and read the assigned studies.

### 6.2 Construct validity

Construct validity refers to the degree to which our study capture and represent the relevant concepts or phenomena of interest using the right tools and tests. We have identified two construct validity threats on our study: *grey literature bias and search bias.*

Initially, we conducted a systematic mapping in search of proposals that assist during modelling. That triggered a *grey literature bias* since our systematic mapping focuses on published literature, excluding relevant information from *grey literature.* Mitigating this threat was a priority since the MDSE field is extremely practical and not all data is published in scientific literature. Therefore, we decided to run a review of the MDSE tools on the market using the GMQ. This allowed us to analyse grey literature, such as documentation, web sites, and user guides. However, we recognize that there is other grey literature that could have been included but were not, such as reports and white papers.

Regarding *search bias validity threat,* we mitigate this threat by using two different search strategies in our systematic mapping study: database search and forward/backward snowballing. Despite this, we recognize that the selection of scientific databases (in this case five different ones) can trigger the search bias validity threat. In addition,

the selection of the initial studies to perform snowballing introduces a search bias. However, we highlight that we have chosen five relevant databases in software engineering and chose a strategy to select the initial studies for snowballing to mitigate this thread. Regarding the selection of MDSE tools, using a single study trigger the *search bias validity threat.* Nevertheless, we consider that GMQ follows a research-based strategy to gather an overview as representative as possible of the MDSE tool market. However, in future work, we consider it relevant to explore other studies or conduct an independent search to find MDSE tools on the market.

### 6.3 External validity

External validity is concerned with whether we can generalise and apply our research findings beyond the specific context and sample used. We have identified a *language bias* in our study. We include only proposals written in English. Then, studies written in other languages are underrepresented. However, we consider our study a representative sample since most of literature in software engineering is written in English.

## 7 Research outlook on modelling assistance in MDSE tools

Up to this point, we have provided a big picture of proposals for assisting users during modelling tasks in MDSE tools both in literature and practice by answering RQ1, RQ2, RQ3, and RQ4. We observe that modelling assistance have attracted both researchers and MDSE practitioners to improve MDSE tooling. Despite that, we have observed in the data collected in RQ2, RQ3, and RQ4 that not all proposals, both in the literature and in practice, provided data about: their limitations, evaluation metrics, and target users. This lack of data hinders comparison between proposals and triggers a need to improve modelling assistance proposals' design.

After reviewing the data obtained in each RQ, we consider relevant to unify the results of our study. we have collected information on the strategies, goals, limitations, target users, and evaluation metrics. In Fig. 11, we summarise the results on these clusters. In section 7.1, we show how researchers and practitioners can take advantage of these clusters and design novel modelling assistants.

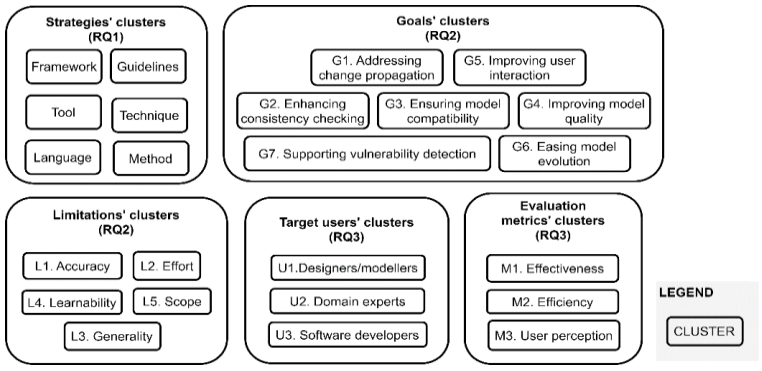


**Fig. 11.** Resulting clusters: strategies, goals, limitations, target users, and evaluation metrics.

### 7.1 AssistMDSE: An emerging framework to design software modelling assistants

In previous research [66], we have collected user requirements on how to assist users during modelling tasks in MDSE tools by using focus groups as requirements elicitation protocol. As a result, we proposed a framework focused on the "who", target users to assist; the “how”, strategy to assist target users; and the “what”, specific set of requirements aim to be met. Taking into account the results from [66] together with the results from the systematic mapping study and analysis on the state of practice presented in sections 4 and 5, we propose AssistMDSE, an emerging framework to design software modelling assistants (see Fig. 12). AssistMDSE supports a clear identification of target users (e.g., designers, modellers, domain experts, software developers) and a more concrete strategy for modelling assistance by categorising assistants into groups like framework, tool, language, guidelines, technique, and method.

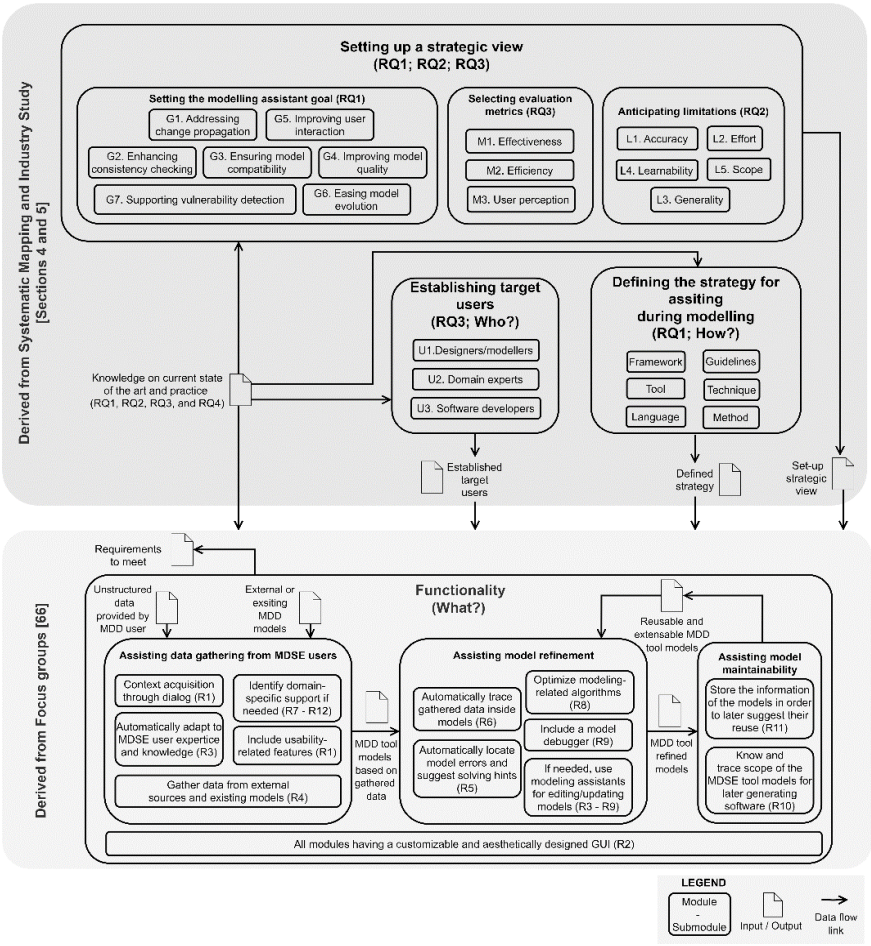



**Fig. 12.** AssistMDSE emerging framework.

### AssistMDSE illustrative case

The AssistMDSE framework is our first step towards a method to design modelling assistants that encompasses both the strategic vision—i.e., goals, limitations, and evaluation metrics—and execution vision provided by the *who*, *how*, and *what.* To illustrate its practical use we introduce the following use case:

**A modelling assistant for automatically creating Digital Health models in MDSE tools.**

**Context:** Whatscount GmbH is a Swiss-young and innovative company in the field of digitalisation that has dedicated itself to creating digital health software using an in-house MDSE tool. The Whatscount modellers noticed they were investing too much time creating models only to realize that these models had already been created before. Therefore, they started to collect information in the shape of ‘modelling patterns’ that

repeated across their various projects, which they could reuse to create new models and decrease modelling time.

**Objective:** The Whatscount modellers need a modelling assistant that help them to formally specify modelling patterns and reuse them in their software development projects.

**Solution:** Use the AssistMDSE framework to design a modelling assistant for the specification and use of modelling patterns (see Table 6).

**Table 6.** Using AssistMDSE to design a modelling assistant modelling patterns specification and reuse.

| Design item | Description | Cluster | Sources keywords |
|---|---|---|---|
| **Setting the modelling assistant goal** | **Create new models** based on previous identified modelling patterns to decrease modelling time, describing digital health applications | **G6. Easing model evolution.** Involves proposals implementing model instances, **creating models**, co-evolving models based on historical data, or editing existing models | Automatically generate models [31, 41, 51, 60] |
| **Selecting evaluation metrics** | The success will be measured as the difference of **modelling time** before and after implementing it | **M2. Efficiency.** Involves proposals measuring the effort, including metrics such as **modelling time,** execution time, and model completion time, among others | Modelling time [55, 61] |
| **Anticipating Limitations** | Create new models only in the **digital health domain**[6] | **L3. Generality.** Involves proposals that their results are limited to specific modelling scenarios or **specific domains** | Limitations on specific domains [45, 62] |
| **Establishing target users (Who?)** | Whatscount GmbH **modellers** are the target users | **U1. Designers/modellers.** Involves proposals that define their target users as persons who **specify models**/designs in MDSE tools | Modellers [39, 45] |
| **Defining the strategy for assisting during modelling (How?)** | **AI-empowered modelling assistant that uses modelling patterns** for creating models in an MDSE tool | **Tool**. Includes **artificial intelligence-empowered software assistants**, modelling environments, and transformation-based tools, among others. | Artificial intelligence-empowered software assistants [33] |
| **Functionality (What?)** | i) **Gather data from modellers** to re-use previous identified modelling patterns in digital health applications; ii) **suggest inputs and options** based on the previous modellers provided information in the digital health domain; and iii) **storage/parse the resulting models** in a MDSE tool | **Assisting data gathering from MDSE users - R1.** Context acquisition through dialog (i). **R7-R12.** Identify domain specific support if needed (ii). **R4.** Gather data from external sources and existing models (iii). | User requirements from [66] |

**Results and conclusions:**

[6] More limitations would arise after implementation/validation phases.

AssistMDSE guided Whatscount modellers to define an initial design of their modelling assistant for creating models in digital health. AssistMDSE has guided Whatscount modelers in defining the initial design of their modelling assistant. Moreover, through AssistMDSE, Whatscount's modellers now have potential related works to compare their proposed solution. This will facilitate the reuse of existing proposals and position their work with them. This initial design serves as the roadmap for continue the modelling assistant development, considering the information established within the framework. For instance, Whatscount modellers' next step in their development process involves creating a mock-up of their modelling assistant to demonstrate its feasibility (see Figure 15). Here, they reflect the decisions made with AssistMDSE: a question-based process for modelers (who?) that they can answer as fast as possible (evaluation metric) to create models (goal) based on digital health patterns (limitation), all empowered by artificial intelligence (how?).

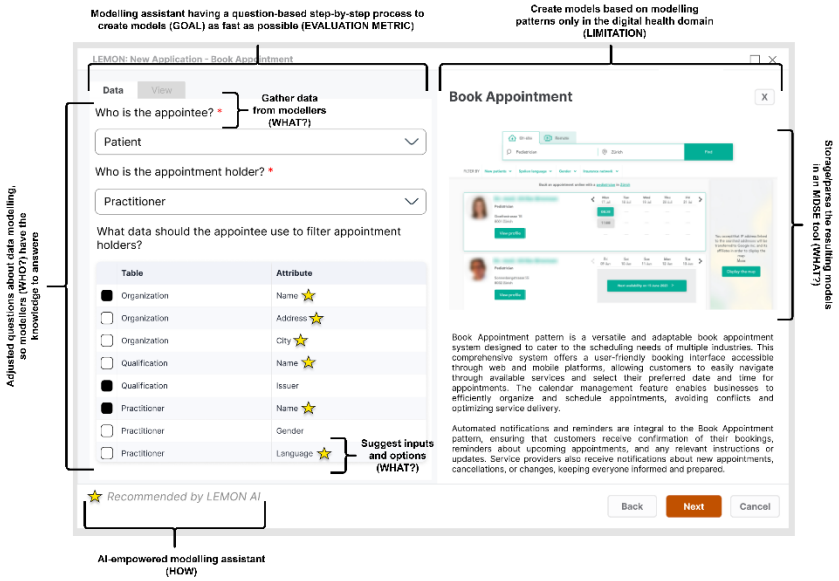



**Fig. 13.** Information specified in AssistMDSE reflected in more advanced stages of the Whatscount' modelling assistant.

## 8 Conclusions and Future work

In this paper, we have presented results around a main research question: what proposals exist in the literature and practice to assist users during modelling tasks in MDSE tools? To do so, we have conducted a systematic mapping study and explore the state of the practice on assisting users during modelling tasks. As a result, three reviewers selected 44 proposals from a set of 2.613 records to answer the following research questions: i) how is software modelling assisted? ii) what goals and limitations existing MDSE tools have? and iii) which evaluation metrics and target users existing MDSE tools consider? As a result, we reported clusters on modelling assistance strategies,

goals, limitations, evaluation metrics, and target users that emerged from the extracted data. Moreover, we have reviewed the documentation on 17 MDSE tools available in practice based on the Gartner Magic Quadrant for enterprise low-code application platforms. Using tools documentation, we extracted data to answer iv) what is the state of the practice on modelling assistance?

After analysing the results of our study (see Section 4 and 5), we have observed that both in the literature and in practice there is interest on assisting users during modelling tasks in MDSE tools. Proof of that is we found 44 studies on literature specifically aiming to assist users during modelling tasks. On the other hand, we observed answering RQ4 that most MDSE tools we reviewed from market have no documented how they intended to assist users during modelling tasks. However, most of the leading MDSE tools based on GMQ have such documentation, showing that leader MDSE tools consider relevant to mention how they assist users during modelling tasks. Furthermore, we observe that MDSE tools in practice mentioned common strategies, goals, limitations, evaluation metrics, and target users with proposals in literature. In overall, this shows the relevance of designing in detail how to assist users during modelling tasks in practice and literature.

Despite the observed significance of assisting during modelling tasks, our data from RQ2, RQ3, and RQ4 reveals a gap in both the literature and market-ready proposals. Specifically, we have observed that information regarding their limitations, evaluation metrics, and target users is either scattered or non-existent, despite having data on goals and strategies for assisting in modelling tasks. This hinders carrying out an objective comparison between proposals. .

In contrast with the results, nowadays we face disruptive new technologies such as artificial intelligence emerging. These edge technologies represent a disruptive shift in the strategies for assisting users during modelling tasks that we have reported in our study (RQ1) that are awaiting to come in the following years. We based our observation having in mind the introduction of Large Language Models (LLMs) and Generative-Pretrained Transformers (GPTs) to assist diverse users on other software development tasks than modelling such as generating code based on natural language [85].

Recognizing the potential for disruptive changes in software modelling assistance strategies motivates providing a shared framework where researchers can query the current modelling assistance strategies (RQ1) and the objectives pursued through these strategies (RQ2). This stands in contrast to the inherent limitations within each proposal (RQ2). By doing so, new technologies driven by artificial intelligence can leverage existing knowledge, avoiding the reinvention of the wheel, and addressing previously identified limitations.

From the non-technical modelling assistance design, we would like to emphasize the significance of explicitly specifying the users who a proposal intends to assist during modelling (RQ3). Previous experiences—such as the one reported on [86]—emphasize that the success of innovations in MDSE development tools is greatly influenced by the users—a.k.a. user profiles—who interact with these tools. Authors state that lacking a definition of the users hindered their expectation management and user interaction fine-tuning process. This is why the lack of explicit definition in the proposals compiled in our study is of concern to us.

In view of these considerations, we believe that a common framework is indispensable for facilitating the design of novel proposals on software modelling assistance in the years to come. Therefore, we introduce AssistMDSE: an emerging framework for designing software modelling assistants. We expect researchers and MDSE practitioners to propose new solutions for modelling assistance in MDSE tools based on our findings and AssistMDSE. Thus, they can align their solutions with already existing goals and target existing gaps, avoid mentioned limitations, consider listed evaluation metrics, and target specific users. Also, consider current solutions in practice. In this paper, we also demonstrate how to take advantage of AssistMDSE by illustrating a case where a company designs a novel modelling assistant for the specification and reuse of modelling patterns in digital health applications.

We recognize that more research is needed to fully utilize the results of our study and realize the full potential of AssistMDSE. Locally, our focus will be on further validation and method engineering efforts to provide a more structured approach for developing new modelling assistants in MDSE. However, we believe that a solo effort would not be sufficient to exploit the potential of our results. That is why we plan to create a public repository based on future versions of AssistMDSE to serve as a knowledge base for researchers and practitioners designing modelling assistants in MDSE tools. The aim of this repository is to connect new modelling assistant designs with related works, identifying similarities and potential synergies. Additionally, it will be a resource for the community to analyse and compare existing proposals. For example, in Section 7.1, we introduced a modelling assistant as an illustrative case. With an existing repository that follows the AssistMDSE structure, we could graphically visualize (see Fig. 14) how such modelling assistant is related to existing works (see Table 6; column “Source keywords”). With the support of the community nurturing this repository with their designs, the modelling assistant connections with existing works can be sustained over time and evolve.

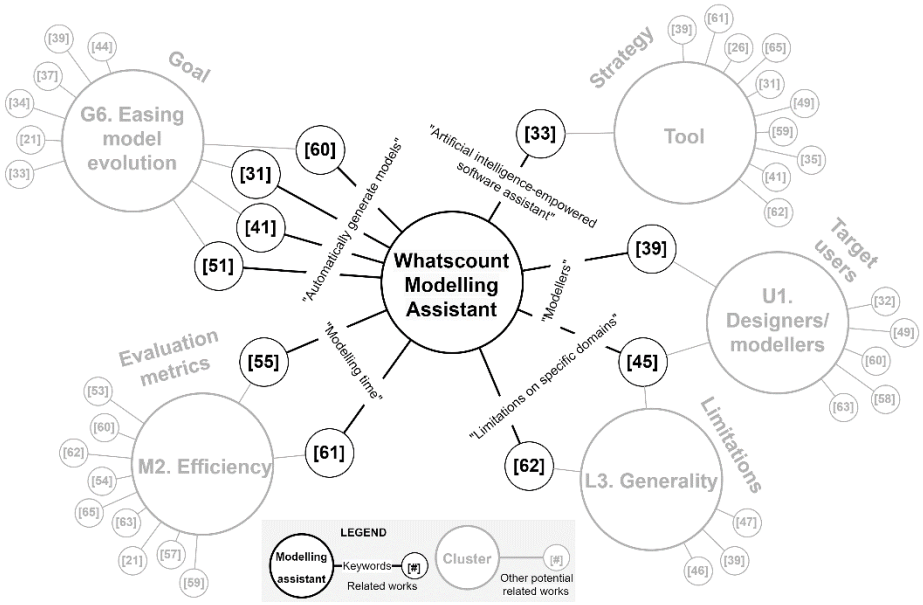



**Fig. 14.** Example of visualization in a public repository for designing modelling assistants based on AssistMDSE clusters.

In conclusion, we underscore the vital role of both local-level and community-level efforts in advancing the landscape of modelling assistance. We reflect our next steps in a research agenda that includes further validation and creating a public repository in Fig. 15.

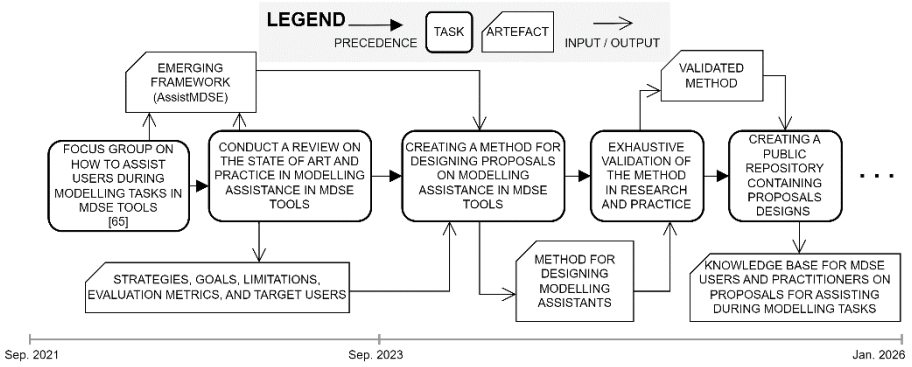



**Fig. 15.** Research agenda.

## Acknowledgements

We would like to express our sincere gratitude to Anastassios Martakos and Flavio Einsenring for their invaluable contribution as reviewers in the systematic mapping protocol. We are truly grateful for their time and dedication during study selection process.

## Competing interest statement

The authors declare no competing financial or non-financial interests that could influence the interpretation of the findings or bias the publication of this research.

## Funding sources

This research was fully supported by the ZHAW Institute for Applied Information Technology (InIT), the Innosuisse Flagship SHIFT project, and the ZHAW School of Engineering.

## References

1. Sendall, S., Kozaczynski, W.: Model transformation: the heart and soul of model-driven software development. IEEE Softw. 20, 42–45 (2003). https://doi.org/10.1109/MS.2003.1231150.
2. Panach, J.I., España, S., Dieste, Ó., Pastor, Ó., Juristo, N.: In search of evidence for model-driven development claims: An experiment on quality, effort, productivity and satisfaction. Inf Softw Technol. 62, 164–186 (2015). https://doi.org/10.1016/j.infsof.2015.02.012.
3. Domingo, Á., Echeverría, J., Pastor, Ó., Cetina, C.: Evaluating the Benefits of Model-Driven Development. In: International Conference on Advanced Information Systems Engineering (CAiSE'20). pp. 353–367 (2020). https://doi.org/10.1007/978-3-030-49435-3_22.
4. Mussbacher, G., Combemale, B., Kienzle, J., Abrahão, S., Ali, H., Bencomo, N., Búr, M., Burgueño, L., Engels, G., Jeanjean, P., Jézéquel, J.-M., Kühn, T., Mosser, S., Sahraoui, H., Syriani, E., Varró, D., Weyssow, M.: Opportunities in intelligent modeling assistance. Softw Syst Model. 19, 1045–1053 (2020). https://doi.org/10.1007/s10270-020-00814-5.
5. Magalhães, A.P., Maciel, R.S.P., Andrade, A.: Developing model transformations: A systematic literature review. In: 22nd International Conference on Enterprise Information Systems. pp. 80–89 (2020). https://doi.org/10.5220/0009380100800089.
6. Iung, A., Carbonell, J., Marchezan, L., Rodrigues, E., Bernardino, M., Basso, F.P., Medeiros, B.: Systematic mapping study on domain-specific language development tools. Empir Softw Eng. 25, 4205–4249 (2020). https://doi.org/10.1007/s10664-020-09872-1.
7. Czech, G., Moser, M., Pichler, J.: Best practices for domain-specific modeling. A systematic mapping study. In: 44th Euromicro Conference on Software Engineering and Advanced Applications. pp. 137–145 (2018). https://doi.org/10.1109/SEAA.2018.00031.
8. Abade, A., Ferrari, F., Lucrédio, D.: Testing M2T transformations: A systematic literature review. In: 17th International Conference on Enterprise Information Systems. pp. 177–187 (2015). https://doi.org/10.5220/0005378501770187.

9. Almonte, L., Guerra, E., Cantador, I., de Lara, J.: Recommender systems in model-driven engineering: A systematic mapping review. Softw Syst Model. 21, 249–280 (2021). https://doi.org/10.1007/s10270-021-00905-x.
10. He, C., Mussbacher, G.: Model-driven engineering and elicitation techniques: A systematic literature review. In: IEEE 24th International Requirements Engineering Conference Workshops. pp. 180–189 (2017). https://doi.org/10.1109/REW.2016.9.
11. Franzago, M., Ruscio, D. Di, Malavolta, I., MucCini, H.: Collaborative model-driven software engineering: A classification framework and a research map. IEEE Transactions on Software Engineering. 44, 1146–1175 (2018). https://doi.org/10.1109/TSE.2017.2755039.
12. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Inf Softw Technol. 64, 1–18 (2015). https://doi.org/10.1016/j.infsof.2015.03.007.
13. Kitchenham, B., Charters, S.M.: Guidelines for performing Systematic Literature Reviews in Software Engineering . , Durham, UK (2007).
14. Belur, J., Tompson, L., Thornton, A., Simon, M.: Interrater Reliability in Systematic Review Methodology: Exploring Variation in Coder Decision-Making. Sociol Methods Res. 50, 837–865 (2021). https://doi.org/10.1177/0049124118799372.
15. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics. 33, 159–174 (1977). https://doi.org/10.2307/2529310.
16. Moody, D.L.: The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods. In: ECIS 2003 Proceedings. pp. 79–96 (2003).
17. Gartner: Gartner Magic Quadrant for Enterprise Low-Code Application Platforms. (2023).
18. Eclipse: Eclipse Modelling Framework, https://www.eclipse.org/modeling/emf/, last accessed 2022/11/29.
19. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: 18th International Conference on Evaluation and Assessment in Software Engineering. pp. 1–10 (2014). https://doi.org/10.1145/2601248.2601268.
20. Ormeño, Y.I., Panach, J.I.: Mapping study about usability requirements elicitation. In: International Conference on Advanced Information Systems Engineering. pp. 672–687 (2013). https://doi.org/10.1007/978-3-642-38709-8_43.
21. Almeida da Silva, M.A., Mougenot, A., Blanc, X., Bendraou, R.: Towards Automated Inconsistency Handling in Design Models. In: International Conference on Advanced Information Systems Engineering (CAiSE'10). pp. 348–362 (2010). https://doi.org/10.1007/978-3-642-13094-6_28.
22. Ilic, D., Troubitsyna, E., Laibinis, L., Leppänen, S.: Formal Verification of Consistency in Model-Driven Development of Distributed Communicating Systems and Communication Protocols. In: 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. pp. 425–432 (2006). https://doi.org/10.1109/ISoLA.2006.40.
23. de Fombelle, G., Blanc, X., Rioux, L., Gervais, M.-P.: Finding a Path to Model Consistency. In: LNCS. pp. 101–112 (2006). https://doi.org/10.1007/11787044_9.
24. Dam, K.H., Winikoff, M.: Generation of Repair Plans for Change Propagation. In: International Workshop on Agent-Oriented Software Engineering. pp. 132–146 (2007). https://doi.org/10.1007/978-3-540-79488-2_10.
25. Sajjad, R., Sarwar, N.: NLP based verification of a UML class model. In: 6th International Conference on Innovative Computing Technology. pp. 30–35 (2016). https://doi.org/10.1109/INTECH.2016.7845070.
26. Kehrer, T., Kelter, U., Ohrndorf, M., Sollbach, T.: Understanding model evolution through semantically lifting model differences with SiLift. In: 28th IEEE International Conference on

Software Maintenance (ICSM). pp. 638–641 (2012). https://doi.org/10.1109/ICSM.2012.6405342.
27. Szabo, C., Chen, Y.: A Model-Driven Approach for Ensuring Change Traceability and Multi-model Consistency. In: 22nd Australian Software Engineering Conference. pp. 127–136. IEEE (2013). https://doi.org/10.1109/ASWEC.2013.24.
28. Ivkovic, I., Kontogiannis, K.: Tracing evolution changes of software artifacts through model synchronization. In: 20th IEEE International Conference on Software Maintenance. pp. 252–261 (2004). https://doi.org/10.1109/ICSM.2004.1357809.
29. Engels, G., Küster, J.M., Heckel, R., Groenewegen, L.: Towards consistency-preserving model evolution. In: International workshop on Principles of software evolution. pp. 129–132. ACM Press, New York, New York, USA (2002). https://doi.org/10.1145/512035.512066.
30. Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M.: From Personal Task Management to End-User Driven Business Process Modeling. In: International Conference on Business Process Management. pp. 84–99 (2008). https://doi.org/10.1007/978-3-540-85758-7_9.
31. Akiki, P.A., Bandara, A.K., Yu, Y.: Cedar studio: an IDE supporting adaptive model-driven user interfaces for enterprise applications. In: Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems - EICS '13. p. 139. ACM Press, New York, New York, USA (2013). https://doi.org/10.1145/2494603.2480332.
32. Elaasar, M., Noyrit, F., Badreddin, O., Gérard, S.: Adaptation and Implementation of the ISO42010 Standard to Software Design and Modeling Tools. In: Communications in Computer and Information Science. pp. 236–258. Springer Verlag (2019). https://doi.org/10.1007/978-3-030-11030-7_11.
33. Savary-Leblanc, M.: Improving MBSE Tools UX with AI-Empowered Software Assistants. In: ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion. pp. 648–652 (2019). https://doi.org/10.1109/MODELS-C.2019.00099.
34. Zander, S., Ahmed, N., Hua, Y.: Empowering the Model-driven Engineering of Robotic Applications using Ontological Semantics and Reasoning. In: 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. pp. 192–198 (2016). https://doi.org/10.5220/0006086201920198.
35. Ben Fraj, I., BenDaly Hlaoui, Y., BenAyed, L.: A reactive system for specifying and running flexible cloud service business processes based on machine learning. In: IEEE 45th Annual Computers, Software, and Applications Conference. pp. 1483–1489 (2021). https://doi.org/10.1109/COMPSAC51774.2021.00220.
36. Ohrndorf, M., Pietsch, C., Kelter, U., Kehrer, T.: ReVision: A Tool for History-based Model Repair Recommendations. In: 40th International Conference on Software Engineering. pp. 105–108 (2018). https://doi.org/10.1145/3183440.3183498.
37. Hadaytullah, Koskimies, K., Systa, T.: Using Model Customization for Variability Management in Service Compositions. In: IEEE International Conference on Web Services. pp. 687–694 (2009). https://doi.org/10.1109/ICWS.2009.92.
38. Almasri, N., Korel, B., Tahat, L.: Verification Approach for Refactoring Transformation Rules of State-Based Models. IEEE Transactions on Software Engineering. 48, 3833–3861 (2021). https://doi.org/10.1109/TSE.2021.3106589.
39. Agt-Rickauer, H., Kutsche, R.-D., Sack, H.: Automated Recommendation of Related Model Elements for Domain Models. In: Communications in Computer and Information Science. pp. 134–158 (2019). https://doi.org/10.1007/978-3-030-11030-7_7.
40. Pérez, F., Valderas, P., Fons, J.: Towards the Involvement of End-Users within Model-Driven Development. In: International Symposium on End User Development. pp. 258–263 (2011). https://doi.org/10.1007/978-3-642-21530-8_23.

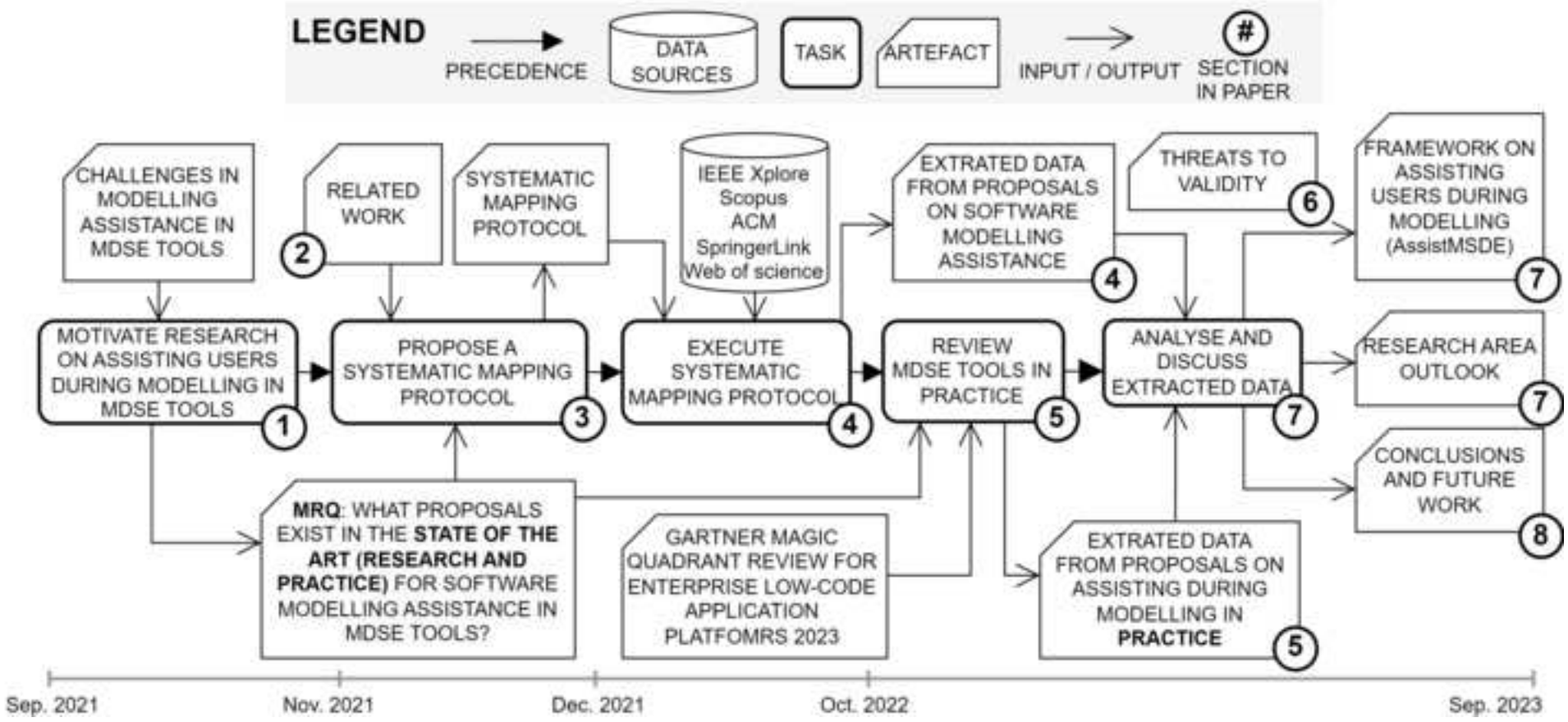
41. Salemi, S., Selamat, A.: Enhancement Approachof Object Constraint Language Generation. J Phys Conf Ser. 933, 1–12 (2018). https://doi.org/10.1088/1742-6596/933/1/012008.
42. Hennig, S., Van den Bergh, J., Luyten, K., Braune, A.: User Driven Evolution of User Interface Models – The FLEPR Approach. In: IFIP Conference on Human-Computer Interaction. pp. 610–627 (2011). https://doi.org/10.1007/978-3-642-23765-2_41.
43. Gogolla, M., Hilken, F., Doan, K.-H.: Achieving model quality through model validation, verification and exploration. Comput Lang Syst Struct. 54, 474–511 (2018). https://doi.org/10.1016/j.cl.2017.10.001.
44. Getir, S., Rindt, M., Kehrer, T.: A Generic Framework for Analyzing Model Co-Evolution. In: ME 2014 - Models and Evolution Workshop Proceedings. pp. 12–21. , Valencia, España (2014).
45. Oberweis, A., Reussner, R.: Model Validation and Verification Options in a Contemporary UML and OCL Analysis Tool. In: MODELLIERUNG'2016. pp. 205–220. , Karlsruhe, Germany (2016).
46. Wang, C., Cavarra, A.: Checking Model Consistency Using Data-Flow Testing. In: 16th Asia-Pacific Software Engineering Conference. pp. 414–421 (2009). https://doi.org/10.1109/APSEC.2009.58.
47. Van Gorp, P., Stenten, H., Mens, T., Demeyer, S.: Towards Automating Source-Consistent UML Refactorings. In: International Conference on the Unified Modeling Language. pp. 144–158 (2003). https://doi.org/10.1007/978-3-540-45221-8_15.
48. Michaux, J., Blanc, X., Shapiro, M., Sutra, P.: A semantically rich approach for collaborative model edition. In: ACM Symposium on Applied Computing. pp. 1470–1475 (2011). https://doi.org/10.1145/1982185.1982500.
49. Oliveira, R., Dingel, J.: Supporting Model Refinement with Equivalence Checking in the Context of Model-Driven Engineering with UML-RT. In: Model-Driven Engineering, Verification and Validation Workshop at the MODELS conference. pp. 1–8 (2017).
50. Di Rocco, J., Di Ruscio, D., Heinz, M., Iovino, L., Lämmel, R., Pierantonio, A.: Consistency Recovery in Interactive Modeling. In: EXE 2017. pp. 1–7 (2017).
51. Schottle, M., Kienzle, J.: Concern-oriented interfaces for model-based reuse of APIs. In: 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS). pp. 286–291. IEEE (2015). https://doi.org/10.1109/MODELS.2015.7338259.
52. Kehrer, T., Kelter, U., Ohrndorf, M., Sollbach, T.: Understanding model evolution through semantically lifting model differences with SiLift. In: 28th IEEE International Conference on Software Maintenance. pp. 638–641 (2012). https://doi.org/10.1109/ICSM.2012.6405342.
53. Wuwei Shen, Kun Wang, Egyed, A.: An Efficient and Scalable Approach to Correct Class Model Refinement. IEEE Transactions on Software Engineering. 35, 515–533 (2009). https://doi.org/10.1109/TSE.2009.26.
54. Steimann, F., Ulke, B.: Generic Model Assist. In: LNCS. pp. 18–34 (2013). https://doi.org/10.1007/978-3-642-41533-3_2.
55. Pourali, P., Atlee, J.M.: A Focus+Context Approach to Alleviate Cognitive Challenges of Editing and Debugging UML Models. In: ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems. pp. 183–193 (2019). https://doi.org/10.1109/MODELS.2019.000-3.
56. Floch, J., Carrez, C., Cieślak, P., Rój, M., Sanders, R.T., Shiaa, M.M.: A comprehensive engineering framework for guaranteeing component compatibility. Journal of Systems and Software. 83, 1759–1779 (2010). https://doi.org/10.1016/j.jss.2010.04.075.

57. Ohrndorf, M., Pietsch, C., Kelter, U., Grunske, L., Kehrer, T.: History-based Model Repair Recommendations. ACM Transactions on Software Engineering and Methodology. 30, 1–46 (2021). https://doi.org/10.1145/3419017.
58. Paz, A., Boussaidi, G. El, Mili, H.: checsdm: A Method for Ensuring Consistency in Heterogeneous Safety-Critical System Design. IEEE Transactions on Software Engineering. 47, 2713–2739 (2021). https://doi.org/10.1109/TSE.2020.2966994.
59. Chavez, H.M., Shen, W., France, R.B., Mechling, B.A., Li, G.: An Approach to Checking Consistency between UML Class Model and Its Java Implementation. IEEE Transactions on Software Engineering. 42, 322–344 (2016). https://doi.org/10.1109/TSE.2015.2488645.
60. Wright, J.M., Dietrich, J.B.: Non-Monotonic Model Completion in Web Application Engineering. In: 2010 21st Australian Software Engineering Conference. pp. 45–54. IEEE (2010). https://doi.org/10.1109/ASWEC.2010.17.
61. Fuhrmann, H., von Hanxleden, R.: Taming Graphical Modeling. In: International Conference on Model Driven Engineering Languages and Systems. pp. 196–210 (2010). https://doi.org/10.1007/978-3-642-16145-2_14.
62. Bürger, J., Jürjens, J., Wenzel, S.: Restoring security of evolving software models using graph transformation. International Journal on Software Tools for Technology Transfer. 17, 267–289 (2015). https://doi.org/10.1007/s10009-014-0364-8.
63. Dam, H.K., Winikoff, M.: An agent-oriented approach to change propagation in software maintenance. Auton Agent Multi Agent Syst. 23, 384–452 (2011). https://doi.org/10.1007/s10458-010-9163-0.
64. Sousa, K., Mendonça, H., Lievyns, A., Vanderdonckt, J.: Getting users involved in aligning their needs with business processes models and systems. Business Process Management Journal. 17, 748–786 (2011). https://doi.org/10.1108/14637151111166178.
65. Kehrer, T., Kelter, U., Taentzer, G.: A rule-based approach to the semantic lifting of model differences in the context of model versioning. In: 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011). pp. 163–172. IEEE (2011). https://doi.org/10.1109/ASE.2011.6100050.
66. Mosquera, D., Ruiz, M., Pastor, O., Spielberger, J.: Assisted-Modeling Requirements for Model-Driven Development Tools. In: International Conference on Research Challenges in Information Science. pp. 458–474 (2022). https://doi.org/10.1007/978-3-031-05760-1_27.
67. Gartner: Gartner Magic Quadrant: Research Method, https://www.gartner.com/en/research/methodologies/magic-quadrants-research, last accessed 2023/05/12.
68. OutSystems: OutSystems low-code platform, https://www.outsystems.com/low-code-platform/, last accessed 2023/05/12.
69. Mendix: Mendix Platform, https://www.mendix.com, last accessed 2023/05/12.
70. Microsoft: Microsoft Power Apps, https://powerapps.microsoft.com, last accessed 2023/05/12.
71. Salesforce: Salesforce platform, https://www.salesforce.com, last accessed 2023/05/12.
72. ServiceNow: ServiceNow platform, https://www.servicenow.com, last accessed 2023/05/12.
73. Oracle: Oracle APEX, https://apex.oracle.com/, last accessed 2023/05/12.
74. Appian: Appian platform, https://appian.com, last accessed 2023/05/12.
75. Zoho: Zoho Creator, https://www.zoho.com/creator/, last accessed 2023/05/12.
76. Pega: Pega Systems, https://www.pega.com/products/platform, last accessed 2023/05/12.
77. Retool: Retool platform, https://retool.com, last accessed 2023/05/12.
78. Newgen: NewgenONE Low Code Digital Transformation Platform, https://newgensoft.com/platform/, last accessed 2023/05/12.
79. Unqork: Unqork platform, https://www.unqork.com, last accessed 2023/05/12.

80. Huawei: Huawei Astro Zero Platform, https://www.huaweicloud.com/intl/en-us/product/appcube.html, last accessed 2023/05/12.
81. Creatio: Creatio ONE platform, https://www.creatio.com, last accessed 2023/05/12.
82. Alibaba: YiDA: Low-code development platform, https://www.alibabacloud.com/en/product/yida, last accessed 2023/05/12.
83. Kintone: Kintone: Rapid Application Development, kintone.com/solutions/rapid-application-development/, last accessed 2023/05/12.
84. Quickbase: Quickbase non-code platform, https://www.quickbase.com/product/product-overview, last accessed 2023/05/12.
85. Cámara, J., Troya, J., Burgueño, L., Vallecillo, A.: On the assessment of generative AI in modeling tasks: an experience report with ChatGPT and UML. Softw Syst Model. 22, 781–793 (2023). https://doi.org/10.1007/s10270-023-01105-5.
86. Daniel, G., Cabot, J.: Applying model-driven engineering to the domain of chatbots: The Xatkit experience. Sci Comput Program. 232, 103032 (2024). https://doi.org/10.1016/j.scico.2023.103032.

Graphical Abstract





**Declaration of interests**

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

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: