IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 September 2022, accepted 4 November 2022, date of publication 7 December 2022,
date of current version 21 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227504

SURVEY

Characterizing UX Evaluation in Software Modeling Tools: A Literature Review

REYHANEH KALANTARI AND TIMOTHY C. LETHBRIDGE, (Senior Member, IEEE)
Department of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada

Corresponding author: Reyhaneh Kalantari (Reyhaneh.kalantari@uottawa.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant 145157.

ABSTRACT Model-Based Software Engineering (MBSE) has a high potential to play a critical role in the whole process of software engineering, bringing many benefits to all stakeholders, yet it is not used by most software developers today, due to both lack of tool capabilities and poor user experience (UX) of tools. This study aims to understand the evaluation types and methods applied by researchers when studying UX in modeling tools (modeling experience or MX) and the types of issues uncovered in these studies. We conducted a literature review using a snowballing approach to gather all studies of this topic. A total of 41 research papers were reviewed. Data extraction was performed based on research questions and a categorization of discussed issues was presented. Several gaps and future opportunities were identified and discussed, which include 1) utilizing interview method in research design; 2) distributing testing tasks based on user profiles; 3) involving UX experts in analysis; 4) scalability testing using large models; 5) assessing MX in areas other than just usability and utility; 6) considering collaborative modeling as an important factor contributing to MX; 7) considering both language issues and tool issues in UX evaluation of software modeling tools; 8) improving the taxonomy of MX challenges; 9) triangulating using multiple methods; and 10) developing and validating MX tool design heuristics.

INDEX TERMS Software modeling, MBSE tool, usability evaluation, user experience.

I. INTRODUCTION

Modeling is a well-researched activity in software engineering. It can bring many benefits by presenting an abstract view of a complex system, simplifying communication among stakeholders, and generating reliable code. Models of software can be used to specify the data, behaviour, architecture and many other system aspects.

Despite the potential value-added of model-centered methodologies, many studies report that they are still not widely used: Lu et al. [1] explored model-based software engineering (MBSE) usage in some Chinese companies in 2018, and the result showed that 45% of respondents do not use an MBSE approach in their teams, and among those who use it, only 3% leverage it in the implementation phase of software development. Petre's study [2] also revealed that 70% of software engineering professionals do not use the Unified Modeling Language (UML), 22% use it only in an informal and selective way, and only 6% use UML for code generation. Another survey result in the embedded system industry states that only 2% of professional respondents *always* use UML, and 11% never use it [3]. Lack of tool support and usability issues have been recognized as recurring factors limiting adoption [4], [5], [6], [7], [8], [9], [10], [11], [12].

There are a variety of software modeling technologies available. As with any type of product, tool users prefer those supporting achievement of their goals in a manner that gives them a feeling of satisfaction. Accordingly, usability and the broader topic of user experience (UX) have become highly relevant in the whole process of software development [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. We will discuss the UX concept in detail in Section II.C., but in brief, it encompasses any aspect of the system that the user experiences, including the features (utility), the usability (how easy it is to learn and use, etc.), its reliability, and similar factors.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao Liu.

Abrahao et al. [14] introduced a new term; MX (Modeling eXperience) to describe user experience in MBSE; they relate many known issues to modeling tools' poor user experience. The authors highlight some challenges and opportunities around this subject and indicate that more studies and empirical research are needed to build a body of knowledge in MX. They also suggest evaluating existing tools as a promising approach to improve the MX of tools.

In this paper, our objective is to present a comprehensive view of the state of the art of MX studies by analyzing current literature; our intended outcome is gaining both practical knowledge in the domain and directions for future research.

We sought to answer the following questions from our study of the literature:

- RQ1: What are the trends and themes of the publication in this field?
- RQ2: What are the characteristics of MBSE Tool Evaluations?
  - RQ2-1: What evaluation methods have been used to study MX?
  - RQ2-2: What data collection methods have been used to study MX?
  - RQ2-3: How many participants are used in different kinds of user evaluations?
  - RQ2-4: What is the profile of participants in user evaluations?
  - RQ2-5: What were the most-used tools and model types in the evaluations?
  - RQ2-6: What is the size of models that have been examined in user testing evaluations?
- RQ3: What are the challenges identified in MBSE tools by publications? What are the most recurring issues reported by studies?

Taken together, answers to these questions should help tool developers improve the quality of tools by considering the identified challenges. The results should also help direct future research in the MX field.

The remainder of this paper is structured as follows. Section II describes the background and definition of concepts. Section III presents the research methodology, research process, and validation criteria. An overview of the results and answers to research questions with defined classifications are given in Section IV. The paper ends with discussions and implications in Section V, research limitations and threats to validity in Section VI, and a conclusion in Section VII.

II. BACKGROUND AND DEFINITIONS

This section explains key terminology used in the paper and summarizes the history of the concepts.

A. MODELING

Models are abstract representations of systems, used to hide details, and decrease the perceived complexity. Models have been used in software development from the early years of computer systems [15], and many practitioners know them as a key factor in achieving software project success, since they empower both engineering and communication aspects of the software process. Holt [16] categorizes the reasons for software projects' failure into three groups: complexity, poor communication, and lack of understanding. He introduces modeling as a solution to mitigate all three of these. This solution is achieved by "1) creating a mental picture of the final system, 2) specifying a system, 3) creating a template as a system plan, and 4) documenting the whole process of system development." Booch [17] defines models as, "a simplification of reality that is created in order to better understand the system under development, as we cannot comprehend complex systems."

A modeling language can be graphical or textual. Graphical modeling languages commonly use nodes as their entities and arcs to represent relationships. ERD (Entity Relationship Diagrams), SysML (System Modeling Language), and UML (Unified Modeling Language) are three examples of graphical software modeling languages, among which UML is the most prominent. "UML is a general-purpose visual modeling language used to specify, visualize, construct and document the artifacts of any system [16]". It was adopted as a standard for the computer industry by the OMG (Object Management Group) in 1997. UML employs a spectrum of diagrams to provide various views of a system, covering static (structural) views and dynamic (behavioral) views [17].

B. MODEL-BASED SOFTWARE ENGINEERING

Model-Based Software Engineering (MBSE) is a software development approach which makes models the central artifacts, rather than code. It is intended to achieve goals such as improving communication among stakeholders, enhancing problem understanding, and increasing productivity of problem solving, typically by generation of some or all of the code, and not just for documentation [18].

Many studies have investigated the advantages of using MBSE and have measured its impacts. Mohagheghi [19] indicates that increasing productivity and improving quality are the ultimate motivations of businesses to leverage the MBSE approach. MBSE should fulfil this by facilitating automation, standardization, formalism, communication, information sharing, early assessment, reuse, and cost estimation.

Luna et al. [9] point out the productivity gains by using modeling tools in software engineering. Barcelona et al. [20] explained the outcome of applying model driven engineering to web development. These gains consisted of shifting the focus more towards the problem and requirements understanding rather than coding, as well as reducing the total effort, time, and cost of development.

Burgueño et al. [22] highlight the necessity of providing a core set of concepts, elements and practices used in this domain.

Savary [10] believes that "applying MBSE methodology is no more a question, and we shall now wonder how to do it rather than if we should." However, there is

a gap between researchers' and practitioners' mindsets, so the MBSE approach is still not widely used among practitioners.

C. USER EXPERIENCE

The term ''User Experience'' (UX) was first introduced by Don Norman in 1993 [23], and the interest of practitioners and researchers in the concept increased when they became convinced that a usability framework focusing only on user performance has limitations [24], since there is a need to satisfy users' growing expectations. UX covers various concepts, from traditional usability (focusing on efficiency of use, learnability, and error prevention) to emotional, experiential, hedonic, and aesthetic variables [25].

Hassenzahl and Tractinsky [25] define UX as, ''consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g., complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g., organizational/social setting, the meaningfulness of the activity, voluntariness of use, etc.).''

Zarour and Alharbi [26], proposed a framework consisting of UX dimensions, aspects categories, and measurement methods. They categorized UX into four dimensions including Value, NX (need experience), BX (brand experience), and TX (technological experience), and then provided corresponding aspects and methods for each dimension. In another study, Zarour [27], developed a user experience needs evaluation method mapping the quality factors introduced in the previous paper with ISO 25000 standard factors for software products, and using evaluation theory. The criteria in his evaluation method are ''usefulness'', ''pleasure'', ''aesthetics'', and ''trust''.

Kumaresh et al. [28], analyses UX of e-commerce platforms based on certain characteristics including home delivery, security and convenient, and flexibility.

Hinderks et al. [29], studied UX management in agile software development using a systematic literature review. They reviewed the approaches and methods used to integrate agile processes with UX practices. Their study highlighted the lack of a common definition of UX management.

Law et al. [24] investigated UX's scope by surveying 275 domain researchers and practitioners and concluded that UX is dynamic, context-based, and inherently subjective, which could be defined by interacting with a product, system, service, or object. As a result of these characteristics, the topic has become controversial, with much effort being devoted to describing and defining its scope in different contexts.

We intend to define and characterize UX in one specific context: the domain of *modeling*. We refer to Hartson and Pyla's definition as our foundation [30]:

''User experience is the totality of the effects felt by the user before, during, and after interaction with a product or system in an ecology.''

Hartson and Pyla present four components for UX: usability, usefulness (utility), emotional impact, and meaningfulness. In our study, we utilize usability, utility, and emotional components; we also add reliability and marketing to cover particularly meaningful issues in the domain of software modeling.

D. USER EXPERIENCE IN MBSE TOOLS

Some studies point to bad user experience or certain factors of user experience as key barriers to adoption of modeling tools [4], [31], [32], [33]. However, we could not find much research focusing on all the relevant aspects of user experience in modeling tools or provide any comprehensive guideline to MX evaluation. As discussed in the previous section, user experience is a broad concept covering a range of issues, which needs a definition based on the context. As far as we know, Abrahao [14] is the first researcher in this domain who used the term MX and emphasizes the need for more theoretical and empirical research to define the user experience in modeling tools. That being said, this paper attempts to gather and analyses existing literature that in some manner has dealt with quality, usability and UX evaluation of modeling tools.

III. METHODOLOGY

This study aims to gain a better understanding about MBSE tool evaluation as well as the known problems and challenges in MX. Therefore, we applied a systematic literature review (SLR) method, whose objective is to evaluate and summarize published information to address issues regarding the subject in an unbiased way. Kitchenham [34] defines a systematic review as a ''means of identifying, evaluating, and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest. Individual studies contributing to a systematic review are called primary studies; a systematic review is a form a secondary study.''

Undertaking an SLR based on searching in databases is common in many disciplines. However, it has some challenges, including the need to formulate good expressions (the key activity of database searches), the different interfaces to the databases, and differing limitations of the databases [27]. These challenges become highlighted when searching in a domain that does not have a standard terminology to use as keywords in the search query, or where the vocabulary uses very general words.

We benefited from the ''guidelines for systematic literature reviews'' provided by Wohlin [35], which describes *snowballing* as a good research approach for SLR in software engineering and formulates the steps of its procedure. Fig. 1 demonstrates the procedure of snowballing.

When applying snowballing, one first identifies a *start set* of papers [35]. With the start set, one then uses backward and forward snowballing. ''Backward snowballing means using the reference list to identify new papers to include, and forward snowballing refers to identifying new papers based on those papers citing the paper being examined. After backward and forward snowballing, new papers identified in the

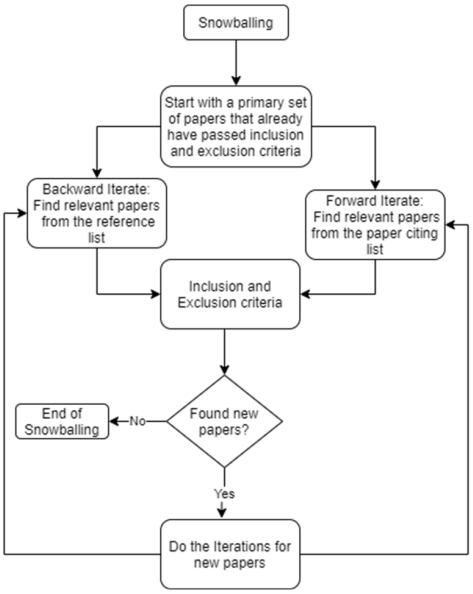

FIGURE 1. Snowballing procedure (based on [35]).

iteration are put into a pile to go into the next iteration'' [35]. The iterations continue until no more new papers are found.

A. SEARCH PROCESS

We used Google Scholar to identify the start set, which Wohlin's [35] recommends as a good choice to avoid publisher bias. Since we are using snowballing, we did not need to craft a comprehensive search query; we only needed to find a few papers that would be among the interconnected set of relevant papers.

We initiated the search process by using the expression below, derived from keywords in our research questions.

''(user experience OR UX) evaluation in (software modeling tools OR MBSE OR MDE)''

This query retrieved around 2 million results in Google Scholar and presented them ordered by the search algorithm's computation of relevance. To further assure relevance, we then manually screened the first 30 papers. For each paper, we first screened based on its title, applying the inclusion and exclusion criteria presented in the next section; for any paper that was not yet excluded, we went further and read its abstract. If we still could not decide about its relevance, we read the whole paper to decide whether to include it.

Following the above screening, we obtained a primary set of six most-relevant papers, which we have marked with an asterisk (∗) in Table 2. Backward and forward snowballing in the first iteration gave 364 results, of which 18 papers were selected to include in our pool. In the second iteration, we found 1183 papers, among which 13 papers were included. The next iteration was performed with 606 papers resulting from snowballing, and we selected 4 more relevant papers considering our inclusion and exclusion criteria. For the last iteration, from 169 papers, we could not find any new relevant papers for our study. Therefore, the snowballing

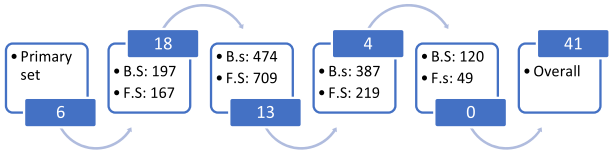

FIGURE 2. Paper counts after each backward (B.S.) plus forward (F.S.) snowballing step.

procedure for this study was finished by finding 41 papers for further analysis. The process is shown in Fig. 2.

B. INCLUSION AND EXCLUSION CRITERIA

We selected references based on the inclusion and exclusion criteria stated in Table 1. The criteria are based on the research goals and questions.

TABLE 1. Inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
Papers in English language	Papers that only present specific feature requirements for creating MBSE tools
Papers published between the years 2000-2020 inclusive.	Papers that present adoption issues beyond MBSE tool usage like modeling language issues
Papers that highlight MBSE tool usage issues by evaluation	Grey literature (thesis, non-refereed papers, manuals, books, and so on)

C. DATA EXTRACTION

We developed a template based on our research questions and entered all relevant information about each resulting paper. The extraction process included the following information:

- Title
- Publication type
- Publication year
- Goal of study
- Focus of evaluation
- Evaluation method
- Data collection tools
- The number and types of assessed tools and diagrams
- The average number of participants in user testing
- Participants' profiles
- Size of evaluated models
- Challenges or issues

IV. RESULTS

This section summarizes and categorizes the findings.

The list of selected papers based on their publication year, publication type, approach, and goal are shown in Table 2. The papers are organized in this and subsequent tables using reverse chronological order. Journal articles are tagged with 'J' in the third column of the table, and Conference articles, including workshops and symposia, are tagged 'C'.

TABLE 2. List of selected papers.

Ref.	Author(s)	Conf./ Journal	Title	Approach	Goal of study
[36]	(Ozkaya & Erata, 2020)	C	Understanding Practitioners' Challenges on Software Modeling: A Survey	W	Understand the challenges that practitioners face
[32]	(Planas & Cabot, 2020)	C	How are UML class diagrams built-in practice? A usability study of two UML tools: Magicdraw and Papyrus	CE	Draw some useful lessons that help to improve the UML modeling process
[37]	(Ren et al., 2020)	C	Collaborative Modelling: Chatbots or On-Line Tools? An Experimental Study	NF	Evaluate the suitability of chatbots for collaborative modeling
[38]	(Bucchiarone et al., 2020)	C	Papyrus for gamers, let's play modeling	NF	Use gamification to help students learning to model
[39]	(Ozkaya, 2019)	C	Are the UML modeling tools powerful enough for practitioners? A literature review	CE	Analyze a set of existing UML modeling tools
*[40]	(Agner et al., 2019)	C	Student experience with software modeling tools	CE	Help professors and students choose tools based on their strengths and weaknesses
*[4]	(Weber et al., 2019)	C	Usability of Development Tools: A CASE-Study	EF	Test and analyze several evaluations methods
[6]	(Stegmaier et al., 2019)	C	Insights for Improving Diagram Editing Gained from an Empirical Study	W	Identify how diagram editing can be improved by investigating how people model on whiteboards
*[10]	(Savary-Leblanc, 2019)	C	Improving MBSE Tools UX with AI-Empowered Software Assistants	NF	Implement software assistants in a modeling tool
[41]	(Badreddin et al., 2018)	C	A Decade of Software Design and Modeling: A Survey to Uncover Trends of the Practice	W	Uncover trends and the adoption patterns of modeling languages such as UML
[9]	(Robles Luna et al., 2018)	C	Challenges for the adoption of model-driven web engineering approaches in industry	W	Present the current problems of MDE approaches
[42]	(Mert Ozkaya & Ferhat Erata, 2018)	C	Analyzing UML Modeling Tools for Practical Use	CE	Assess tools' support for requirements
[43]	(Pietron et al., 2018)	C	A Study Design Template for Identifying Usability Issues in Graphical Modeling Tools	EF	Present a study design template for identifying usability issues in graphical editors
*[44]	(Pourali, 2018)	C	An Empirical Investigation to Understand the Difficulties and Challenges of Sofware Modellers When Using Modelling Tools	W	Identify the most prominent difficulties that users might face when using UML modeling tools
[45]	(Störrle, 2018)	C	Improving model usability and utility by layered diagrams	NF	Understand how layered diagrams are useful in modeling tools
[46]	(Agt-Rickauer et al., 2018)	C	DoMoRe – A Recommender System for Domain Modeling	NF	Support the domain modeling process by implementing an automated modeling recommendation
[3]	(Akdur et al., 2018)	J	A survey on modeling and model-driven engineering practices in the embedded software industry	W	Investigate practices in the embedded software engineering projects
[12]	(Liebel et al., 2018)	J	Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice	W	Assess the current state-of-practice and the challenges of embedded systems domain are facing due to shortcomings with MBE
[33]	(Agner & Lethbridge, 2017)	C	A Survey of Tool Use in Modeling Education	CE	Evaluate tool support for teaching modeling
[8]	(Whittle et al., 2017)	J	A taxonomy of tool-related issues affecting the adoption of model-driven engineering	W	Present a taxonomy of tool-related considerations
[31]	(Vogelsang et al., 2017)	C	Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry	W	Investigate issues of MBSE adoption in embedded software developing companies
[11]	(Bordeleau et al., 2017)	C	Challenges and Research Directions for Successfully Applying MBE Tools in Practice	W	Discuss challenges in applying MBE from academic and industrial viewpoints
[47]	(Ribeiro et al., 2016)	C	Comparative analysis of workbenches to support DSMLs: Discussion with non-trivial Model-Driven Development needs	CE	Analysis and comparison of three tools

TABLE 2. *(Continued.)* **List of selected papers.**

Ref.	Author(s)	Conf./ Journal	Title	Approach	Goal of study
*[48]	(Safdar et al., 2015)	C	Empirical Evaluation of UML Modeling Tools– A Controlled Experiment	CE	Compare the productivity of the software engineers while modeling with the tools
[49]	(Ahmar et al., 2015)	C	Enhancing the communication value of UML models with graphical layers	NF	Using the layers in UML diagram editors
[50]	(Ferreira et al., 2014)	C	Characterizing the Tool-notation-people Triplet in Software Modeling Tasks	EF	Present a usability evaluation template and evaluate two tools based on that
[51]	(Williams et al., 2014)	C	Software Analytics for MDE Communities	EF	Propose to apply software analytics techniques on open source MDE tools and languages
[52]	(Whittle et al., 2013)	C	Industrial Adoption of Model-Based Engineering: Are the Tools Really the Problem?	W	Present a taxonomy of tool-related issues
[53]	(Condori-Fernández et al., 2013)	J	An empirical approach for evaluating the usability of model-driven tools	EF	Present a usability evaluation template
[54]	(Kuhn et al., 2012)	C	An Exploratory Study of Forces and Frictions Affecting Large-Scale Model-Driven Development	W	Investigate model-driven engineering
[55]	(El Kouhen et al., 2012)	C	Evaluation of Modeling Tools Adaptation	CE	Review tool's adaptation approaches
[56]	(Heena & Ranjna, 2011)	C	A comparative study of UML tools	CE	Compare most-used tools based on some features
[57]	(Panach et al., 2011)	C	An Experimental Usability Evaluation Framework for Model-Driven Tools	EF	Present a usability evaluation template
[5]	(Huang et al., 2011)	C	Hammering Models: Designing Usable Modeling Tools	EF	Facilitate improvement of modeling tools by identifying common problems in existing tools and developing a framework of redesign guidelines
[58]	(Eichelberger et al., 2009)	C	A Comprehensive Survey of UML Compliance in Current Modelling Tools	CE	Asses the UML compliance levels of modeling tools
[59]	(Saraiva & Silva, 2008)	C	Evaluation of MDE Tools from a Metamodeling Perspective	EF	Present an evaluation framework for tool support of the metamodeling activity, and evaluate a small set of tools
*[60]	(Auer et al., 2007)	C	Explorative UML modeling comparing the usability of UML tools	CE	Assess tools' usability for UML sketching
[61]	(Störrle, 2007)	C	Large Scale Modeling Efforts: A Survey on Challenges and Best Practices	W	Present challenges in large modeling
[62]	(Egyed, 2006)	C	Instant consistency checking for the UML	NF	Introduce an approach for quickly, correctly, and automatically deciding when to evaluate consistency rules in modeling
[7]	(Lahtinen & Peltonen, 2005)	C	Adding speech recognition support to UML tools	NF	Present an approach to develop speech interfaces in UML tools
[63]	(Seffah & Rilling, 2001)	C	Investigating the relationship between usability and conceptual gaps for human-centric CASE tools	W	Highlight common usability problems in the most popular Java IDEs

We identified several primary approaches taken by the papers, as follows: The code in brackets at the start of each item, is used to mark the tool in the 'Approach' Column of Table 2.

- (EF) To propose an evaluation framework or template, usually alongside a case study to validate that framework.
- (NF) Proposing a new feature for improving MBSE tools and evaluating tools to validate the suitability of the feature.
- (CE) To comparatively evaluate several tools.
- (W) Presenting the current state and trend of MBSE tools, regarding their weaknesses and adoption issues.

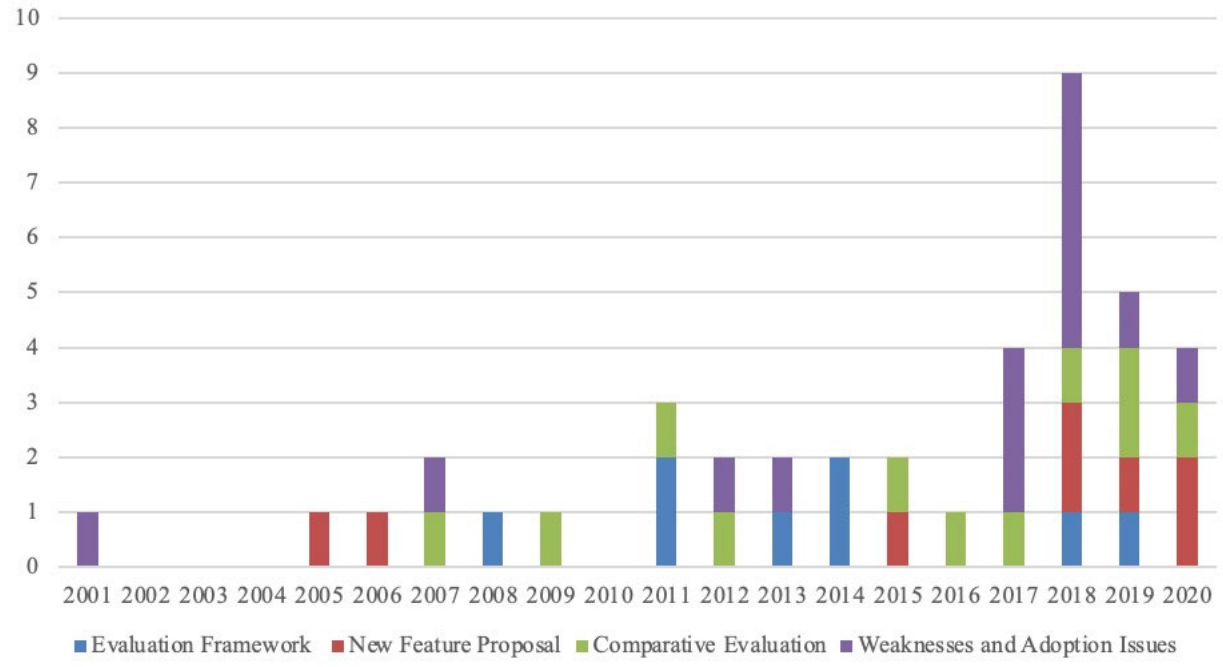

FIGURE 3. Paper publishing trend.

A. RQ1—WHAT ARE THE TRENDS AND THEMES OF THE PUBLICATIONS IN THIS FIELD?

The bar chart in Fig. 3 illustrates the number of published papers over a period of 20 years, in different categories.

An overview of selected papers shows that the number of papers rises after 2016 and reaches the highest point in 2018, with nine papers. The decreasing trend after 2018 is probably due to the delay in publishing papers in the most recent years, and the fact that backward snowballing to recent papers is less likely than it is to earlier papers. It is expected to continue rising in the future, especially because usability and user experience concepts are emerging fields that attract more and more attention from academia and industry.

As the figure shows, most papers, including the oldest one, belong to the category W (Weaknesses). This highlights that investigating weaknesses and adoption issues of MBSE tools, is among the most recurring concerns of researchers in this domain.

B. RQ2—WHAT ARE THE CHARACTERISTICS OF MBSE TOOL EVALUATIONS?

We describe the characteristics of each evaluation using the set of criteria that form the column headings of Table 3. These criteria are the a) focus of the evaluation, b) the data collection methods used (inquiry, inspection and testing), c) the background of tool-user participants when such participants were involved (students, professors, modelers in industry, or a combination of these groups), d) the number of MBSE tools which are assessed by the paper, e) whether the paper lists challenges, f) whether it suggests improvements, and g) the specific suggested improvements (if any).

The focus of evaluation in some cases was usability criteria such as efficiency; in other cases, it was assessing certain requirements, or participants' behavior and experience. Two studies assessed more than 50 tools using the inspection method performed by researchers [39], [58]. As illustrated in Table 3, 85% of papers list some challenges and issues, while 51% suggest improvements, either theoretically or by implementing a particular feature including gamification [38], AI-empowered software assistants [10], layered diagrams [45], [47], instant consistency checking [62], and speech recognition [7].

1) RQ2-1: WHAT EVALUATION METHODS HAVE BEEN USED TO STUDY MX?

Evaluation methods in the reviewed publications covered a wide range of methodologies with different combinations, some of which are more concerned with certain features and functionality of tools, while others emphasize user-centered evaluations.

From 41 papers, 38 papers mentioned their methods explicitly. We categorized these papers based on their employed evaluation method. As shown in Fig. 4, more than half of the publications (58%) applied user-centered evaluation, with the user types shown in the 'Tool-user participant types' column of Table 3. For many other papers (34%) it was the researchers themselves that performed the evaluations, mostly based on feature checklists (with the value 'Inspection' of the 'Data Collection Methods' column of Table 3). Only two papers (5%) leveraged both researcher and tool-user evaluation methods, and just one paper [4], benefitted from UX expert feedback besides user evaluation, and it

was concerned with presenting different usability evaluation methods.

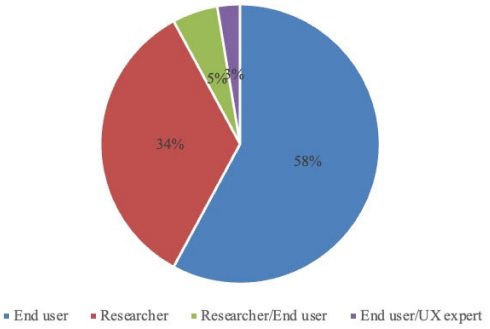

FIGURE 4. Distribution of the types of people using/studying the tool when doing the evaluations in the papers.

2) RQ2-2: WHAT DATA COLLECTION METHODS HAVE BEEN USED TO STUDY MX?

We consider three general types for data collection methods: inspection, inquiry, and testing. Inspection refers to a set of methods that are all based on evaluators feedback about an interface, and includes methods such as heuristic evaluation, cognitive walkthrough, and feature inspection [64]. This method introduced as a more affordable alternative of usability testing, as it does not require participants [65]. Inquiry methods involve user inquiries through interviews or surveys. Testing or user testing methods are a group of methods that are used to gather information from users during their interaction with the product, such as observation, think-aloud, and performance measurements.

Those studies that performed evaluation based on users, employed either the user testing or the inquiry method, and studies performed by researchers usually leveraged the inspection method. There were also some papers that used a combination of them.

As shown in Fig. 5, the most-used data collection method is inquiry, which includes performing surveys by questionnaires or interviews. The inquiry method was used either as the only evaluation method or in combination with other methods. 42% of the papers used the inspection method in their evaluation. Another common approach employed by studies was performing evaluations based on both user testing and inquiry to complement the results, which has shaped 19% of papers. In total, 32% of papers performed some user testing in their data collection methods. More details of data collection instruments will be discussed below.

Looking closer at the applied data collection methods and tools, one can see that some papers (30%) benefited from more than one data collection approach. This helps researchers to achieve a more reliable result and address more issues and challenges.

The following is a list of data collection tools used by our selected papers that typically involve direct interaction with users:

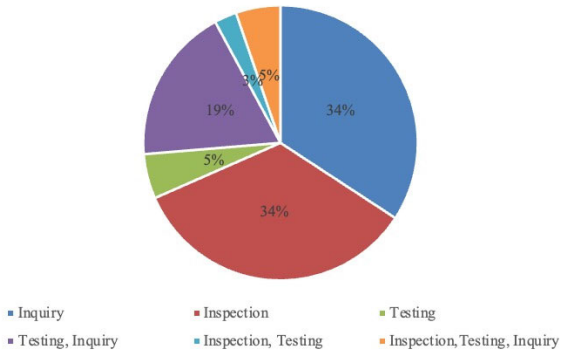

FIGURE 5. Distribution of general categories of data collection methods employed in the papers.

a: QUESTIONNAIRES

This data collection method, which is the most common one in our papers [3], [7], [9], [12], [33], [36], [37], [40], [41], [43], [44], [48], [51] is one of the most popular methods to gather both quantitative and qualitative user data. It is used either in the pre-evaluation phase to gather demographic information of participants or during the evaluation process for gathering the comments and insights of users about the tool. The SUS (System Usability Scale) [43] and the CSUQ (Computer System Usability Questionnaire) [44] are two examples applied in the selected papers. QUIS, SUMI, UMUX and UMUX-LITEare other examples of standard questionnaires in this field [65].

b: INTERVIEWS

This method, which is the second most-used method for user evaluation in our findings [4], [6], [8], [31], [43], [52], [54], [61], [63] is similar to the questionnaire but allows evaluators to add or change some questions based on the context.

c: TASK-BASED

This method, as a quantitative approach to usability testing, is performed by giving users tasks and calculating various metrics by researchers [4], [32], [37], [44], [48], [53], [57]. The metrics include completion rate and error rate, both for measuring the effectiveness of the tools, as well as elapsed time and number of clicks for measuring the efficiency of the tools. To gather the required information, evaluators usually use screen recording, video recording, or interaction logging. There was one paper [48] that gathered information based on users' self-reports.

d: THINK-ALOUD

In this method [4], [5], [43], [44], participants are supposed to speak while interacting with a tool and describe what they expect and what happens instead. As a qualitative method, this helps evaluators better understand the users' thoughts and needs. Videotaping, eye tracking [65], audio recording, or just taking notes by researchers are the complementary instruments used for performing this method.

TABLE 3. Selected papers and their properties.

Ref.	Author(s)	Focus of evaluation	Data collection methods	Tool-user participant types	Number of tools assessed	List challenges	Improvement suggestions	Suggested improvement feature
[36]	Ozkaya & Erata	Eight categories of software modeling challenges	Inquiry	Industry		Yes		
[32]	Planas & Cabot	Efficiency	Inspection, Testing	Academia	2	Yes	Yes	
[37]	Ren et al.	Efficiency, effectiveness, satisfaction	Testing, Inquiry	Academia	2			
[38]	Bucchiarone et al.	-			1	Yes	Yes	Gamification
[39]	Ozkaya	Specific requirements	Inspection		58	Yes	Yes	
[40]	Agner et al.	Learnability	Inquiry	Academia	31	Yes	Yes	
[4]	Weber et al.	Usability criteria	Testing, Inquiry, Inspection	Industry	1			
[6]	Stegmaier et al.	Behavior of users	Testing, Inquiry	Academia		Yes	Yes	
[10]	Savary-Leblanc	-				Yes	Yes	AI-Empowered Software Assistants
[41]	Badreddin et al.	Practitioners' experience	Inquiry	Multi experience level		Yes		
[9]	Robles Luna et al.	Specific requirements	Inquiry	Industry		Yes		
[42]	Ozkaya & Erata	Specific requirements	Inspection		11	Yes		
[43]	Pietron et al.	Usability of graphical editors	Testing, Inquiry	Experts	1			
[44]	Pourali	Efficiency, effectiveness, satisfaction	Testing, Inquiry	Academia Industry		Yes	Yes	
[45]	Störrle	Validation of a new feature	Testing, Inquiry	Academia Industry	1	Yes	Yes	Layered diagrams
[46]	Agt-Rickauer et al.	Validation of a new feature		Industry		Yes	Yes	Automated modeling recommendation
[3]	Akdur et al.	Latest trends in modeling	Inquiry	Academia Industry	10	Yes	Yes	
[12]	Liebel et al.	Practitioners' experience	Inquiry	Industry	3	Yes		
[33]	Agner & Lethbridge	Professors' experience in teaching using modeling tools	Inquiry	Academia	32	Yes	Yes	
[8]	Whittle et al.	Practitioners' experience	Inquiry	Industry		Yes	Yes	
[31]	Vogelsang et al.	Practitioners' experience	Inquiry	Industry		Yes		
[11]	Bordeleau et al.	Experience in industry	Inspection			Yes		
[47]	Ribeiro et al.	Suitability for developing non-trivial language	Inspection		3	Yes		
[48]	Safdar et al.	Productivity (completeness, memory load, learnability, and modeling effort)	Testing, Inquiry	Academia	3	Yes		
[49]	Ahmar et al.	Validation of new feature(layers)	Inspection		1	Yes	Yes	Add layers
[50]	Ferreira et al.	HCI perspectives: 1. Usability 2. Communicability	Inspection		1	Yes		
[51]	Williams et al.	MDE community	Inspection		10			
[52]	Whittle et al.	Practitioners' experience	Inquiry	Industry		Yes	Yes	
[53]	Condori-Fernández et al.	Efficiency, effectiveness, satisfaction	Testing	Multi experience level	1	Yes		
[54]	Kuhn et al.	Practitioners' experience	Inquiry	Industry		Yes		

TABLE 3. *(Continued.)* **Selected papers and their properties.**

Ref.	Author(s)	Focus of evaluation	Data collection methods	Tool-user participant types	Number of tools assessed	List challenges	Improvement suggestions	Suggested improvement feature
[55]	El Kouhen et al.	Customizability and usability	Inspection		5			
[56]	Heena & Ranjna	Specific requirements	Inspection		6	Yes	Yes	
[57]	Panach et al.	Effectiveness and efficiency	Testing	Multi experience level	1	Yes	Yes	
[5]	Huang et al.	Behavior and experience of users	Inspection, Testing, Inquiry	Industry	1	Yes	Yes	
[58]	Eichelberger et al.	Specific requirements	Inspection		68	Yes		
[59]	Saraiva & Silva	Metamodeling activity	Inspection		4	Yes	Yes	
[60]	Auer et al.	Efficiency, effectiveness	Inspection		2	Yes	Yes	
[61]	Störrle	Practitioners' experience	Inquiry	Industry		Yes		
[62]	Egyed	Validation of a new feature	Inspection		1	Yes	Yes	Instant consistency checking
[7]	Lahtinen & Peltonen	Behavior and experience of users	Testing, Inquiry	Novices	1	Yes	Yes	Speech recognition
[63]	Seffah & Rilling	Ease of use	Inquiry	Industry	5	Yes	Yes	

e: USER OBSERVATION

In this method [6], [7], [45], evaluators take notes as they watch users interacting with the tool. It could provide some immediate user feedback and is usually used combined with other methods. This method usually utilizes user recording, such as audio recording and videotaping, to enable researchers of interpreting users' behavior after the study. Fernandez [53] used videotaping to evaluate users' satisfaction based on their facial reactions.

f: FOCUS GROUP

This is a qualitative method to explore people's attitudes. It can be used to find out what issues are of most concern for a community or group. This method was only used by one publication [4], which tried to compare different evaluation methods.

g: PAPER PROTOTYPING

This method involves hand-drawn representations of what the product looks like and is usually used in early design iterations of a product development process, which is efficient in cost and time. Huang et al. [5] employed this method for one of their design iteration processes and stated that, ''the paper prototype is not merely an evaluations tool. However, a material for designers to convey their design concepts as well as a platform for communication among domain experts, engineers, and practitioners''.

The following are the additional data gathering tools and methods that do not involve direct user interaction and are applied by researchers and authors of the papers:

h: FEATURE-BASED EVALUATION

This is the most-used method [39], [42], [47], [55], [56], [59], [62] among all the methods that did not involve user interaction. This is typically done by rating tools against a list of criteria.

i: KLM-GOMS

This method predicts how long it will take an expert user to accomplish a routine task without errors using an interactive computer system. It is a useful method to compare the usability of several tools. Two papers [60] and [66] applied this method in a comparative study.

j: USAGE SCENARIOS

This is a qualitative method [49], [50] for early usability evaluation. A usage scenario describes a system's behavior as it responds to requests from an actor who wants to achieve a particular goal, thorough of which usability issues could be highlighted.

k: HEURISTIC EVALUATION

In this method [4], [5], a group of usability specialists judge and rate products based on a pre-defined set of usability principles.

l: SOFTWARE ANALYTIC

This method was used by a comparative study [51] to evaluate different open-source modeling tools based on their communities.

The bar chart in Fig. 6 compares different data collection tools employed by our selected papers, color coded according to whether data is gathered from researcher's own analysis, consultation with external experts, or interaction with tool users. The questionnaire is the most-used tool in user evaluations; it is one of the most time- and cost-effective techniques.

This result is also aligned with [13], which presents the questionnaire as the most-used usability evaluation method in software engineering.

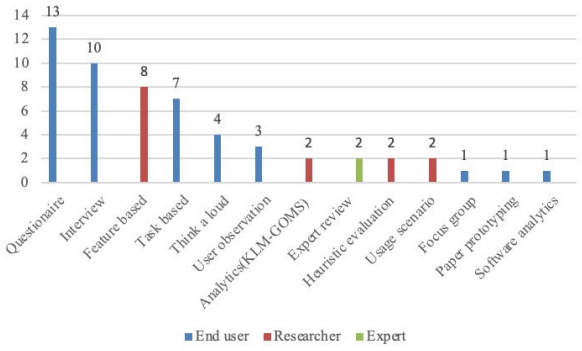

FIGURE 6. The number of papers in our study that employed each data collection method.

3) RQ2-3: HOW MANY PARTICIPANTS ARE USED IN DIFFERENT KINDS OF USER EVALUATION?

The number of participants recruited for an evaluation process is another item worth considering, as it could be a vital factor in the result's reliability. We calculated the average number of participants and categorized them based on the papers' data collection approach. Pietron et al. [43], in their proposed evaluation template, recommend at least 9-15 participants for performing user testing studies. As we can see in the bar chart, the average number of participants in the selected papers is in the mentioned range.

The results show that papers using questionnaires involve the highest number of participants, with an average of 115. This outcome is not surprising as the questionnaire is one of the most efficient methods in terms of time and cost.

In contrast, the user observation data collection approach has the lowest average number of participants (10). This method needs careful attention to detail while recording users and analyzing the qualitative data, so it might not always be affordable to perform with a large number of participants.

As illustrated by Fig. 7, a boxplot diagram is used to represent the distribution of the number of participants. The distance between the average and median value of the *Questionnaire* method shows some outliers in data. Taking a closer look at those outliers reveals that there are a few studies that conducted remote evaluations [33], [40], [41], and therefore contain noticeably more participants than other studies; among which, [3] has the highest amount with 627 participants.

4) RQ2-4: WHAT IS THE PROFILE OF PARTICIPANTS IN USER EVALUATION?

We analyzed the demographic information from 26 papers that applied tool user evaluation.

Illustrated by Fig. 8, in the user testing method, participants should ideally be selected among the real users of the system. Pietron [43] suggests using the TA-EG questionnaire

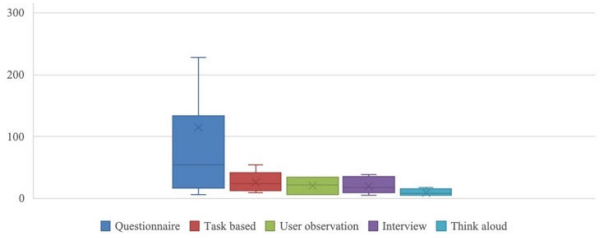

FIGURE 7. The distribution of the number of participants for each data collection approach.

to ensure that the participants have a positive attitude towards technology. The background of participants depends on the target users of the system. Hence, if a tool is supposed to be used by students, researchers, and also practitioners, the best approach is to recruit participants from both academia and industry.

We identified four groups based on the participants' profiles and categorized the papers based on these groups. The industry group covered 46% of papers, including professionals and practitioners mostly from IT-based companies. 19% of papers performed their evaluation with students (undergraduate and graduate students). 19% of papers did not mention participants' background while stating they recruited participants based on different levels of experience [41], [53], [57]. One paper claimed using *expert* users without defining the term [43], and one paper [7] used novice users, which they all grouped under *Experience based*. The number of papers involving both students and industry participants was only 11%. There was only one paper [33] that considered professors who teach modeling as research participants.

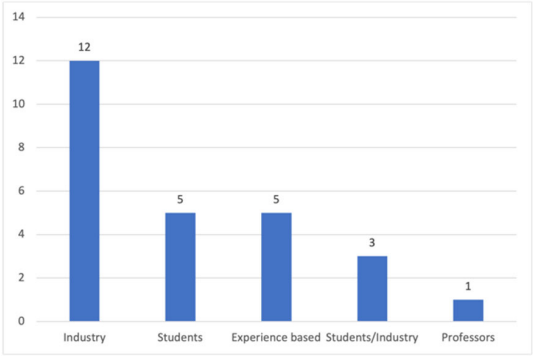

FIGURE 8. Participant profile in papers.

5) RQ2-5: WHAT WERE THE MOST-USED TOOLS AND MODEL TYPES IN THE EVALUATIONS?

More than 60 modeling tools exist in the market, including commercial and open-source tools [67]. The researcher's preference for choosing the tools to be evaluated is an interesting point to consider. In our analysis, we found almost 50% of papers evaluated just one tool, whereas the remaining evaluated several tools and conducted a comparative analysis.

A variety of tools were highlighted by selected papers, some of which were intentionally selected by researchers to

be evaluated, and several tools were mentioned as preferred by users in performed surveys.

In this paper we focused on the most-used tools and disregarded those that were mentioned just once. Fig. 9 shows tools based on their occurrence in different time-period categories; we used three different time periods, since various tools that were commonly evaluated more than a decade ago are no longer widely used, yet other tools have only appeared recently. As the trend in Fig. 9 displays, Papyrus has been a prominent tool from 2016, following by Visual Paradigm, Magic Draw, Enterprise Architect, Eclipse based modeling tools, Argo UML, Umple, Star UML, IBM Rational Rhapsody, and BoUML.

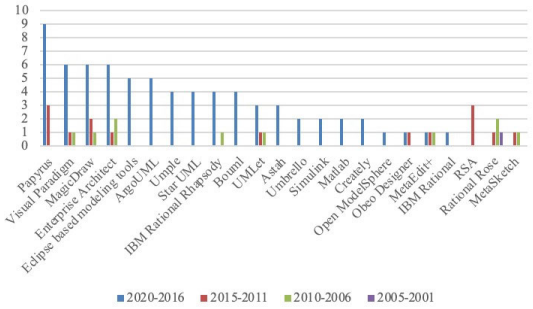

FIGURE 9. The most-used tools in the studied papers.

Papyrus is an open-source MBSE tool with a fully customizable environment that can be adopted for various purposes [68]. It is known to be a distinguished tool by researchers and practitioners; thus, it is not surprising that it was more frequently used in studies than other tools.

Although there are 14 diagrams in UML, only a few of are employed in practice and in evaluation studies. For example, Pietron [43] suggested state machine diagram for the study design template due to its familiarity with students and developers; This allows participants from academia and industry to participate in the evaluation.

Fig. 10 shows that class diagram and state machine diagram are the most popular diagrams used in the evaluations. As the representatives of the system's static and dynamic views, these diagrams can provide a comprehensive view of the system. Following these, sequence diagram, activity diagram, and use case diagram are other diagrams used in the tool evaluation process of our selected papers. As shown by Fig. 10, in total, 12 papers cited particular diagrams in the evaluation process, four of which used more than one diagram, and the remainder used only one diagram. Other papers (29 papers) evaluated programs with no regard for the diagrams.

6) RQ2-6: WHAT IS THE SIZE OF MODELS THAT HAVE BEEN EXAMINED IN USER TESTING EVALUATIONS?

The ability to construct and maintain large, complex models has been considered a critical feature of modeling tools. However, very few studies addressed this topic.

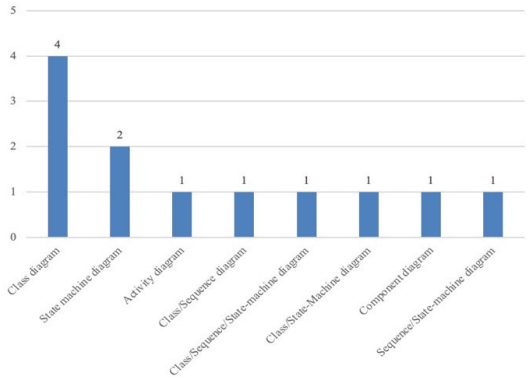

FIGURE 10. Diagram types used in the analyzed papers.

Storrle [61] is one of the researchers in this domain who investigated the challenges of large modeling by interviewing some domain experts. There were several significant challenges identified in this study, such as the need for version control and the problems related to release and deployment; however, these difficulties are not described in depth. In another study, he presented multi-layer feature to mitigate large models' complexity [45]. Yet, it is validated only from the usability aspect, while problems regarding release and deployment are not considered.

Other papers performed tool evaluation using only simple tasks and small models, and it seems that the real practicality of tools is neglected in evaluations. Therefore, it is imperative to consider this aspect of modeling for further studies.

C. RQ3: WHAT ARE THE CHALLENGES IDENTIFIED IN MBSE TOOLS BY PUBLICATIONS?

Among all selected papers, 38 papers address particular challenges or issues, either by evaluations or literature reviews.

To gain an overview of existing known challenges, we needed some standard terms and classification since papers have described their identified issues using different terms.

We chose to categorize the issues into the following categories: Utility, Usability, Reliability, Emotional, and Marketing. We attempt to present issues in distinctive groups, although there are undoubtedly some overlaps.

As [30] defines, Utility "is about the power and functionality of the backend software that gives you the ability to get work (or play) done. It's the real underlying reason for a product or system". To better clarify the gaps, utility issues were further categorized in two subgroups: utility issues in the tools being evaluated and utility issues in the languages supported by the tools.

According to [30], Usability includes factors such as "ease of use", "user performance", "efficiency", "error avoidance", "learnability", and "retainability (ease of remembering)". Similar to utility, usability issues were further categorized in two subgroups: usability of the tool and usability of the modeling language.

Emotional issues, which is the least considered part of UX in literature, includes personal feelings during the usage

of system such as fun, enjoyment, pleasure, aesthetics and so on [30].

The Reliability category was used to contain problems related to system bugs.

Marketing issues fit in a group focusing on how easy it is to obtain, install, and update required features of a system. It also considers the availability and quality of the system communities, which could play a critical role in MBSE tools succeeding.

Table 4 elaborates on each category with certain cases from studies.

After reviewing and categorizing all the issues, we counted items in each group based on their occurrence. If a paper points to one or several items in a specific group, it still counts as one occurrence for this group.

The hierarchy and proportion by each category and subcategory of issues, is illustrated in Fig. 11, highlighting that most repeated issues by papers are in the ''Utility'' and ''Usability'' groups, with a focus on tool issues, whereas ''Emotional'' and ''Reliability'' group with the fewest reported issues.

V. DISCUSSIONS AND IMPLICATIONS

This paper has presented a systematic literature review of usability evaluation methods and has identified challenges for software modeling tools. Snowballing was applied to finding papers starting from a seed set that was determined from a search. Overall, 41 papers were selected to review. Extracted data were categorized and described based on several research questions, through which various limitations and research gaps were identified. These are discussed below.

The following are research gaps that we recommend be considered by future researchers who are studying, improving, or comparing specific MX quality:

A. BENEFIT FROM THE INTERVIEW METHOD IN RESEARCH DESIGN

Our results have shown that the interview method provides challenges in a wider variety of categories than other methods. 50% of papers that present the most issues (from three or more issue categories) have utilized interviews in their research design. The interview method is even more significant in the UX-focused category than the utility category.

B. DISTRIBUTE TESTING TASKS BASED ON USER PROFILES

In terms of sample selection in the user-testing evaluation method, representing the full spectrum of targeted users is important. Our findings indicate that the majority of papers recruited their participants from academia or industry. Future researchers need to consider a broader user distribution, including by level of experience in the domain and expertise in using the system. As Lewis and Sauro say, ''To determine who will participate in the test, the administrator needs to obtain or develop a user profile'' [65]. It is important for evaluators to establish a user profile and its characteristics to increase the representativeness of participants, and hence the validity of studies.

C. INVOLVE UX EXPERTS IN ANALYSIS

Based on our results, only one paper [4] benefited from the input of a UX expert. Our analysis also shows that the papers have the fewest improvement suggestions in the ''Human factors'' group. Consequently, we require more empirical studies in the evaluation of MBSE tools that incorporate UX expert feedback, alongside user studies, in order to gain an overall view and provide reliable suggestions for improving the user experience.

D. TEST FOR SCALABILITY USING LARGE MODELS

Many software systems are very large, yet the literature focuses on evaluation with small models, there is a need for more studies regarding scalability as a critical feature for modeling tool practically.

E. ASSESS MX IN AREAS OTHER THAN JUST USABILITY AND UTILITY

According to the selected papers, there has been a gap in studies concerning emotional factors and marketing practices, which highlights the necessity to carry out more empirical research in this area. In recent years, marketing issues have received increased attention, as app stores and easy installability and updatability have become essential factors influencing awareness and popularity. More studies are required to identify all the impactful factors of these types.

F. CONSIDER COLLABORATIVE MODELING AS AN IMPORTANT FACTOR CONTRIBUTING TO MX

While we categorized collaborative modeling issues as a utility category, it is worth noting that it has a broader effect, as it could impact user performance (usability) as well as user enjoyment (emotional). To facilitate effective team collaboration, modeling tools should provide good support for versioning, model diff/merge, model review, and document generation [11]. Therefore, a significant portion of utility issues require improvement to contribute to collaborative modeling support.

G. CONSIDER LANGUAGE ISSUES AS WELL AS TOOL ISSUES

In most cases, the reported issues are related to the modeling tools used, but language quality plays an equally important role in the overall user experience. Not differentiating between language and tool issues, could cause misleading results in the evaluation process. Considering this, Ferreira [50] provided a conceptual tool to address tools issues using tool-notation-people triplets, which covers all kinds of issues. That paper suggested evaluation methods in each category but does not state any sub-category or criteria for using it as a guideline for tool evaluations. The present paper tries to solve these deficiencies by providing detailed categories and sub-categories in MDE UX evaluation.

TABLE 4. MX issues highlighted in each paper.

Paper	MX Issues
(Ozkaya & Erata, 2020) Understanding Practitioners' Challenges on Software Modeling: A Survey	**Utility (tool):** Analyzing models, Transforming models, Managing models, Separation of concerns **Utility (language):** Extendibility, Developing formal modeling languages **Usability (language):** Complexity
(Planas & Cabot, 2020) How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus	**Usability (tool):** Difficult in use, Lack of a proper guidance **Reliability:** Bugginess
(Ren et al., 2020) Collaborative Modelling: Chatbots or On-Line Tools? An Experimental Study	**Utility (tool):** Collaborative modeling feature
(Bucchiarone et al., 2020) Papyrus for gamers, let's play modeling	**Usability (tool):** Complexity
(Ozkaya, 2019) Are the UML modelling tools powerful enough for practitioners? A literature review	**Utility (tool):** Multiple viewpoints, Scalability, Formal verification, Round-trip engineering, Exportation, Collaborative modeling, Scripting, Project management **Usability (tool):** Complexity
(Agner et al., 2019) Student experience with software modeling tools	**Utility (tool):** Feedback, Interaction with other tools **Usability (tool):** Complexity, Slowness, Difficult to use **Reliability:** Bugginess
(Stegmaier et al., 2019) Insights for Improving Diagram Editing Gained from an Empirical Study	**Utility (tool):** Free sketching feature, highlighting feature, template feature **Usability (tool):** User preference in texts, shapes, sizes, colors, Undo functionality **Reliability:** bugginess
(Savary-Leblanc, 2019) Improving MBSE Tools UX with AI-Empowered Software Assistants	**Usability (tool):** Complexity
(Badreddin et al., 2018) A Decade of Software Design and Modeling: A Survey to Uncover Trends of the Practice	**Utility (tool):** Executable artifact, Collaboration and Communication **Usability (tool):** Learning curve, Complexity
(Robles Luna et al., 2018) Challenges for the adoption of model-driven web engineering approaches in industry	**Utility (tool):** Metamodel support, Traceability and Debuggability, Monitoring technologies, Architecture modeling, Technological aspects modeling such as DB connection **Marketing:** Community and tutorials, Expensive licensing, Close source which leads to lack of control,
(Mert Ozkaya & Ferhat Erata, 2018) Analysing UML Modeling Tools for Practical Use	**Utility (tool):** Various viewpoints, Sub-diagramming for the connectors (e.g., class associations), Formal verification, Collaboration support
(Pourali, 2018) An Empirical Investigation to Understand the Dificulties and Challenges of Sofware Modellers When Using Modelling Tools	**Usability (tool):** Retainability, Error avoidance
(Störrle, 2018) Improving model usability and utility by layered diagrams	**Utility (tool):** Layers feature **Usability (tool):** Complexity
(Agt-Rickauer et al., 2018) DoMoRe – A Recommender System for Domain Modeling	**Usability (tool):** Knowledge intensive
(Akdur et al., 2018) A survey on modeling and model-driven engineering practices in the embedded software industry	**Utility (tool):** Synchronization between software artifacts, Version management, Integration of legacy code, Model checking, Traceability and compatibility, Code generation **Usability (tool):** Need training
(Liebel et al., 2018) Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice	**Utility (tool):** Interoperability between MBE tools, Variability management, Version management **Usability (tool):** Need training
(Agner & Lethbridge, 2017) A Survey of Tool Use in Modeling Education	**Utility (tool):** Interaction with other tools, Supporting modeling aspects **Usability (tool):** Complexity
(Whittle et al., 2017) A taxonomy of tool-related issues affecting the adoption of model-driven engineering	**Utility (tool):** Model analysis, Scalability, Refactoring models, Sketching models, Tool versioning, Flexibility, Customizability, Integration, Migration **Usability (tool):** Complexity, Not match human mental model, Productivity **Usability (language):** Complexity **Emotional:** Trust

TABLE 4. *(Continued.)* **MX issues highlighted in each paper.**

	Marketing: Cost of tools, Sustainability
(Vogelsang et al., 2017) Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry	**Utility (tool):** Incompatibility, Traceability, Modularization, Testing **Usability (tool):** Learnability, Complexity **Marketing:** ROI uncertainty
(Bordeleau et al., 2017) Challenges and Research Directions for Successfully Applying MBE Tools in Practice	**Utility (tool):** Collaboration facilities and customization, Graphical and textual support in a tool, Model diff/merge, Model review, Document generation. **Usability (tool):** Usability issues
(Ribeiro et al., 2016) Comparative analysis of workbenches to support DSMLs: Discussion with non-trivial Model-Driven Development needs	**Utility (tool):** Validation Support, Transformation support **Usability (tool):** Learnability issues
(Safdar et al., 2015) Empirical Evaluation of UML Modeling Tools–A Controlled Experiment	**Usability (tool):** Learnability issues, Productivity issues
(Ahmar et al., 2015) Enhancing the communication value of UML models with graphical layers	**Usability (tool):** Readability issues
(Ferreira et al., 2014) Characterizing the Tool-notation-people Triplet in Software Modeling Tasks	**Utility (tool):** Visibility, Partial semantic verification **Usability (tool):** Hard mental operation Reliability: Error proneness
(Williams et al., 2014) Software Analytics for MDE Communities	**Marketing:** Lack of rich community
(Whittle et al., 2013) Industrial Adoption of Model-Based Engineering: Are the Tools Really the Problem?	**Usability (tool):** HCI concerns **Marketing:** Lack of communities
(Condori-Fernández et al., 2013) An empirical approach for evaluating the usability of model-driven tools	**Utility (tool):** Customizability **Usability (tool):** Novice users are not guided, Interfaces are not visually consistent, Some interfaces provide too much information with regard to the available space, Error messages do not help the user to solve the mistake, Icons are not self-explicative, Some interface titles are confusing, The tool does not provide undo and redo facilities, Some elements do not provide a menu when the user clicks with the right button of the mouse, Some interfaces are obtrusive (they do not allow showing the window below), Some functions are only reachable by means of icons, but not through the menu
(Kuhn et al., 2012) An Exploratory Study of Forces and Frictions Affecting Large-Scale Model-Driven Development	**Utility (tool):** Model diffing, Point-to-point traceability, **Utility (language):** Lack of problem-specific visual "little languages" **Usability (tool):** Long build-cycles prevent live modeling
(Heena & Ranjna, 2011) A comparative study of UML tools	**Utility (tool):** Round-trip engineering, All UML diagrams, UML 2.0, Document generation
(Panach et al., 2011) An Experimental Usability Evaluation Framework for Model-Driven Tools	**Utility (tool):** Compatibility issues **Usability (tool):** Guidance, Error management, Consistency and Adaptability violations, User control
(Huang et al., 2011) Hammering Models: Designing Usable Modeling Tools	**Utility (tool):** Shortcomings in terms of supported functionality **Usability (tool):** Not support different levels of expertise
(Eichelberger et al., 2009) A Comprehensive Survey of UML Compliance in Current Modelling Tools	**Utility (tool):** Compatibility problems
(Saraiva & Silva, 2008) Evaluation of MDE Tools from a Metamodeling Perspective	**Utility (tool):** Model transformations **Utility (language):** Support for language specification (syntax and semantics), **Usability (tool):** Complexity
(Auer et al., 2007) Explorative UML modeling comparing the usability of UML tools	**Usability (tool):** Complexity

TABLE 4. *(Continued.)* **MX issues highlighted in each paper.**

(Störrle, 2007) Large Scale Modeling Efforts: A Survey on Challenges and Best Practices	**Utility (tool):** Version control, Modularity, Migration, Releases, Backups and deployment
(Egyed, 2006) Instant consistency checking for the UML	**Utility (tool):** Consistency checking
(Lahtinen & Peltonen, 2005) Adding speech recognition support to UML tools	**Usability (tool):** Complexity
(Seffah & Rilling, 2001) Investigating the relationship between usability and conceptual gaps for human-centric CASE tools	**Usability (tool):** Interface personalization, Retainability, Error avoidance, Speaking the developer's language, Keeping the developer informed the IDE status

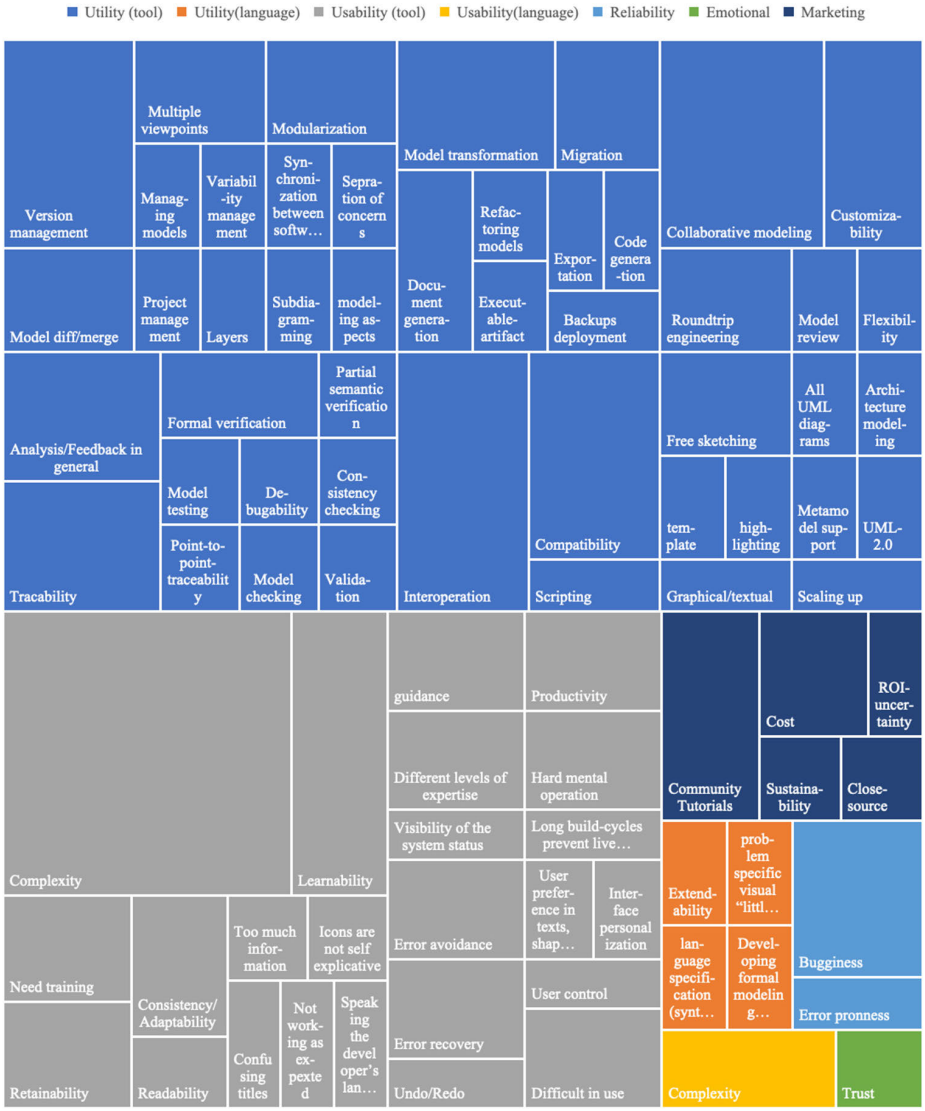

FIGURE 11. **Issue category proportions as a tree map.**

Achieving a comprehensive set of guidelines that consider both tool and language factors in MX, should be a goal.

VI. THREATS TO VALIDITY

As with any review, this study has certain considerations that may limit the soundness of the conclusions drawn.

These biases have been divided into three kinds of threats to validity, which are described below:

Construct validity refers to identifying correct measures for the concept being studied. Zhou [69] pointed out that ''Inappropriate or incomplete search terms in automatic search'', ''Inappropriate inclusion and exclusion criteria'', and ''Restricted time span'' are the most recurring threats to construct validity in SLRs in the domain of software engineering. To mitigate these threats we applied snowballing, and all decisions were checked and rechecked to resolve any inconsistency. We also developed a study protocol during the planning phase, which was reviewed by an external reviewer.

Internal validity refers to the credibility of a causal relationship in a study. Most common threats to internal validity in SLRs are ''Bias in study selection'' [69], ''Misclassification of primary study'' [69] and ''Bias in data extraction'' [69]. We attempt to alleviate these threats by following our peer-reviewed protocol (described in section III), and accurately extracting each paper's data. We used spreadsheets to keep records, and the classification was discussed and reviewed by the authors in a few iterations. Although single-person screening and data extraction could lead to risk of mistakes and inconsistencies, following an unambiguous peer-reviewed protocol makes this approach valid and more feasible in terms of cost and time. Since this paper is part of a thesis research program, the selection process was mostly performed by one author and checked and approved by another author and an external reviewer. In thesis research work, conducting literature reviews with one reviewer is a common practice, however, it may compromise the internal validity of the study. As a result, this is our primary study limitation. We provided all data extraction details in Table 2, Table 3, and Table 4 to allow the reader to verify the reliability and accuracy of the information extracted.

External validity refers to the generalizability of the findings. Excluding papers written in a language other than English and papers with a publication year before 2000 are the main factors may affect the generalization of the results but are common limitations in systematic reviews.

VII. CONCLUSION

There is a need for further research at the meta level, to provide general guidance to MX researchers.

A. IMPROVED TAXONOMY

Researchers and practitioners need a standard classification to use during evaluation. We have attempted in this paper to make further progress in this direction.

As UX issues are complex and interdependent, splitting them in different groups or sub-groups is not always straightforward. Our categorization was performed based on the authors' experience and reviewed in several iterations; however, it is not unexpected to notice some overlaps in group items.

There have been a few categorizations proposed in the literature, but their focuses were on tool adoption issues, and not UX issues, which could be different in certain aspects. Whittle et al. [8] proposed a taxonomy with the focus of determining the factors that prevent organizations from adopting MDE tools. Their study categorized issues in ''technical'', ''organizational'', and ''social'' groups with several subgroups. What seems to be disregarded in that taxonomy, is that human factors go beyond usability into issues such as emotional factors.

Our proposed categorization has been derived based on the issues reported in the papers and may not reflect all the criteria needed to evaluate tools, so future studies should try to extend evaluations to consider aspects covered in literature about other types of software. The resulting taxonomy should then be reviewed iteratively by experts and validated through case studies.

B. TRIANGULATE USING MULTIPLE METHODS

Our results show that most of the papers (73%) performed evaluation by just one of the inspections, inquiry, or testing methods. Only one paper performed evaluation by all three methods, so we cannot claim any conclusions about whether triangulation provides better results. However, since triangulation is a recommended approach by several researchers [70], [71], [72], [73], we suggest further investigation to see if different methods provide different results.

C. COMPREHENSIVE MX DESIGN HEURISTICS

Practitioners also need a comprehensive set of heuristics for improving modeling tools; these can be based on the taxonomy.

Ultimately, what practitioners and tool developers need, is a prioritization of factors and elements based on their relevance in various contexts and with different types of users.

Our study shows that a significant number of papers make suggestions for improvement (see Table. 3). These suggestions fall into two groups: first are the papers that suggest improvements corresponding to their reported issues, and the second group consists of papers who suggest and implement a new feature as a means to resolve an issue, and then evaluate the tool based on such an implementation. Analyzing these suggestions based on their effectiveness, their generalizability, involved tools and evaluation methods, could uncover promising information and brighten a future path in the field.

In conclusion, there are many opportunities to expand, apply and deepen the results of this study, any of which should be employed to brighten the path to a better model-based software engineering tools with an improved UX.

REFERENCES

[1] J. Lu, Y. Wen, Q. Liu, D. Gürdür, and M. Törngren, ''MBSE applicability analysis in Chinese industry,'' in *Proc. INCOSE Int. Symp.*, 2018, vol. 28, no. 1, pp. 1037–1051, doi: 10.1002/j.2334-5837.2018.00532.x.

[2] M. Petre, ''UML in practice,'' in *Proc. 35th Int. Conf. Softw. Eng. (ICSE)*, San Francisco, CA, USA, May 2013, pp. 722–731, doi: 10.1109/ICSE.2013.6606618.

[3] D. Akdur, V. Garousi, and O. Demirörs, ''A survey on modeling and model-driven engineering practices in the embedded software industry,'' *J. Syst. Archit.*, vol. 91, pp. 62–82, Nov. 2018, doi: 10.1016/j.sysarc.2018.09.007.

[4] T. Weber, A. Zoitl, and H. HuBmann, "Usability of development tools: A CASE-study," in *Proc. ACM/IEEE 22nd Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Sep. 2019, pp. 228–235, doi: 10.1109/MODELS-C.2019.00037.
[5] K.-H. Huang, N. Nunes, L. Nobrega, L. Constantine, and M. Chen, "Hammering models: Designing usable modeling tools," in *Proc. IFIP Conf. Hum.-Comput. Interact.*, vol. 6948, Sep. 2011, pp. 537–554, doi: 10.1007/978-3-642-23765-2_37.
[6] M. Stegmaier, A. Raschke, M. Tichy, E.-M. MeBner, S. Hajian, and A. Feldengut, "Insights for improving diagram editing gained from an empirical study," in *Proc. ACM/IEEE 22nd Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Sep. 2019, pp. 405–412, doi: 10.1109/MODELS-C.2019.00063.
[7] S. Lahtinen and J. Peltonen, "Adding speech recognition support to UML tools," *J. Vis. Lang. Comput.*, vol. 16, nos. 1–2, pp. 85–118, Feb. 2005, doi: 10.1016/j.jvlc.2004.08.001.
[8] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, "A taxonomy of tool-related issues affecting the adoption of model-driven engineering," *Softw. Syst. Model.*, vol. 16, no. 2, pp. 313–331, 2017, doi: 10.1007/s10270-015-0487-8.
[9] E. R. Luna, J. M. S. Begines, J. M. Rivero, L. Morales, J. G. Enríquez, and G. H. Rossi, "Challenges for the adoption of model-driven web engineering approaches in industry," *J. Web Eng.*, vol. 17, nos. 3–4, pp. 183–205, 2018, doi: 10.5555/3370055.3370057.
[10] M. Savary-Leblanc, "Improving MBSE tools UX with AI-empowered software assistants," in *Proc. ACM/IEEE 22nd Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Sep. 2019, pp. 648–652, doi: 10.1109/MODELS-C.2019.00099.
[11] F. Bordeleau, G. Liebel, A. Raschke, G. Stieglbauer, and M. Tichy, "Challenges and research directions for successfully applying MBE tools in practice," in *Proc. MDETOOLS*, 2017, p. 6.
[12] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Model-based engineering in the embedded systems domain: An industrial survey on the state-of-practice," *Softw. Syst. Model.*, vol. 17, no. 1, pp. 91–113, Feb. 2018, doi: 10.1007/s10270-016-0523-3.
[13] F. Paz and J. A. Pow-Sang, "A systematic mapping review of usability evaluation methods for software development process," *Int. J. Softw. Eng. Appl.*, vol. 10, no. 1, pp. 165–178, Jan. 2016, doi: 10.14257/ijseia.2016.10.1.16.
[14] S. Abrahao, F. Bourdeleau, B. Cheng, S. Kokaly, R. Paige, H. Stoerrle, and J. Whittle, "User experience for model-driven engineering: Challenges and future directions," in *Proc. ACM/IEEE 20th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS)*, Austin, TX, USA, Sep. 2017, pp. 229–236, doi: 10.1109/MODELS.2017.5.
[15] C. E. Dickerson and D. Mavris, "A brief history of models and model based systems engineering and the case for relational orientation," *IEEE Syst. J.*, vol. 7, no. 4, pp. 581–592, Dec. 2013, doi: 10.1109/JSYST.2013.2253034.
[16] J. Holt, *UML for Systems Engineering: Watching the Wheels*. Edison, NJ, USA: IET, 2004.
[17] G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, J. Connallen, and K. A. Houston, *Object-Oriented Analysis and Design With Applications*, vol. 33, 3rd ed. Reading, MA, USA: Addison-Wesley, 2008. Accessed: Nov. 29, 2020, doi: 10.1145/1402521.1413138.
[18] R. Jolak. (2020). *Understanding and Supporting Software Design in Model-Based Software Engineering*. Accessed: Oct. 18, 2020. [Online]. Available: https://gupea.ub.gu.se/handle/2077/63039
[19] P. Mohagheghi and V. Dehlen, "Where is the proof?—A review of experiences from applying MDE in industry," in *Model Driven Architecture—Foundations and Applications*. Berlin, Germany: Springer, 2008, pp. 432–443, doi: 10.1007/978-3-540-69100-6_31.
[20] M. Barcelona, L. García-Borgoñón, I. Ramos, and M. J. Escalona, "Applying a model-based methodology to develop web-based systems of systems," *J. Web Eng.*, vol. 16, pp. 212–227, Jun. 2017.
[21] M.-J. Escalona, N. Koch, and L. Garcia-Borgoñon, "Lean requirements traceability automation enabled by model-driven engineering," *PeerJ Comput. Sci.*, vol. 8, p. e817, Jan. 2022, doi: 10.7717/peerj-cs.817.
[22] L. Burgueño, F. Ciccozzi, M. Famelis, G. Kappel, L. Lambers, S. Mosser, R. F. Paige, A. Pierantonio, A. Rensink, R. Salay, G. Taentzer, A. Vallecillo, and M. Wimmer, "Contents for a model-based software engineering body of knowledge," *Softw. Syst. Model.*, vol. 18, no. 6, pp. 3193–3205, Dec. 2019, doi: 10.1007/s10270-019-00746-9.
[23] Nielsen Norman Group. *A 100-Year View of User Experience (by Jakob Nielsen)*. Accessed: Dec. 6, 2020. [Online]. Available: https://www.nngroup.com/articles/100-years-ux/
[24] E. L.-C. Law, V. Roto, M. Hassenzahl, A. P. O. S. Vermeeren, and J. Kort, "Understanding, scoping and defining user experience: A survey approach," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2009, pp. 719–728, doi: 10.1145/1518701.1518813.
[25] M. Hassenzahl and N. Tractinsky, "User experience—A research agenda," *Behav. Inf. Technol.*, vol. 25, no. 2, pp. 91–97, Mar. 2006, doi: 10.1080/01449290500330331.
[26] M. Zarour and M. Alharbi, "User experience framework that combines aspects, dimensions, and measurement methods," *Cogent Eng.*, vol. 4, no. 1, Jan. 2017, Art. no. 1421006, doi: 10.1080/23311916.2017.1421006.
[27] M. Zarour, "A rigorous user needs experience evaluation method based on software quality standards," *TELKOMNIKA, Telecommun. Comput. Electron. Control*, vol. 18, no. 5, p. 2787, Oct. 2020, doi: 10.12928/telkomnika.v18i5.16061.
[28] S. Kumaresh, R. Haran, and M. M. Jarret, "Analytics of E-commerce platforms based on user-experience (UX)," in *Intelligent Computing and Innovation on Data Science*. Singapore: Springer, 2021, pp. 309–318, doi: 10.1007/978-981-16-3153-5_34.
[29] A. Hinderks, F. J. Domínguez Mayo, J. Thomaschewski, and M. J. Escalona, "Approaches to manage the user experience process in agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 150, Oct. 2022, Art. no. 106957, doi: 10.1016/j.infsof.2022.106957.
[30] R. Hartson and P. Pyla, "What are UX and UX design?" in *The UX Book*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 3–25, doi: 10.1016/B978-0-12-805342-3.00001-1.
[31] A. Vogelsang, T. Amorim, F. Pudlitz, P. Gersing, and J. Philipps, "Should I stay or should I Go? On forces that drive and prevent MBSE adoption in the embedded systems industry," in *Product-Focused Software Process Improvement*. Cham, Switzerland: Springer, 2017, pp. 182–198, doi: 10.1007/978-3-319-69926-4_14.
[32] E. Planas and J. Cabot, "How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus," *Comput. Standards Interfaces*, vol. 67, pp. 1–13, Jan. 2020, doi: 10.1016/j.csi.2019.103363.
[33] L. T. W. Agner and T. C. Lethbridge, "A survey of tool use in modeling education," in *Proc. ACM/IEEE 20th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS)*, Sep. 2017, pp. 303–311, doi: 10.1109/MODELS.2017.1.
[34] B. Kitchenham, "Procedures for performing systematic reviews," Softw. Eng. Group Dept. Comput. Sci., Keele Univ., Keele, U.K., Tech. Rep., 0400011T.1, Aug. 2004.
[35] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. 18th Int. Conf. Eval. Assessment Softw. Eng.*, London, U.K., 2014, pp. 1–10, doi: 10.1145/2601248.2601268.
[36] M. Ozkaya and F. Erata, "Understanding practitioners' challenges on software modeling: A survey," *J. Comput. Lang.*, vol. 58, Jun. 2020, Art. no. 100963, doi: 10.1016/j.cola.2020.100963.
[37] R. Ren, J. W. Castro, A. Santos, S. Pérez-Soler, S. T. Acuña, and J. de Lara, "Collaborative modelling: Chatbots or on-line tools? An experimental study," in *Proc. Eval. Assessment Softw. Eng.*, New York, NY, USA, Apr. 2020, pp. 260–269. Accessed: Nov. 21, 2020, doi: 10.1145/3383219.3383246.
[38] A. Bucchiarone, M. Savary-Leblanc, X. L. Pallec, J.-M. Bruel, A. Cicchetti, J. Cabot, S. Gerard, H. Aslam, A. Marconi, and M. Perillo, "Papyrus for gamers, let's play modeling," in *Proc. 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst., Companion*, Oct. 2020, pp. 1–5, doi: 10.1145/3417990.3422002.
[39] M. Ozkaya, "Are the UML modelling tools powerful enough for practitioners? A literature review," *IET Softw.*, vol. 13, no. 5, pp. 338–354, Oct. 2019, doi: 10.1049/iet-sen.2018.5409.
[40] L. T. W. Agner, T. C. Lethbridge, and I. W. Soares, "Student experience with software modeling tools," *Softw. Syst. Model.*, vol. 18, no. 5, pp. 3025–3047, Oct. 2019, doi: 10.1007/s10270-018-00709-6.
[41] O. Badreddin, R. Khandoker, A. Forward, O. Masmali, and T. C. Lethbridge, "A decade of software design and modeling: A survey to uncover trends of the practice," in *Proc. 21st ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2018, pp. 245–255, doi: 10.1145/3239372.3239389.
[42] M. Ozkaya and F. Erata. (2018). *Analysing UML Modeling Tools for Practical Use*. Accessed: Nov. 21, 2020. [Online]. Available: https://www.semanticscholar.org/paper/Analysing-UML-Modeling-Tools-for-Practical-Use-Ozkaya-Erata/403b07e2c44bddf034dae5330dd8922122f38410

[43] J. Pietron, A. Raschke, M. Stegmaier, M. Tichy, and E. Rukzio, "A study design template for identifying usability issues in graphical modeling tools," in *Proc. 2nd Workshop Tools Model Driven Eng.*, vol. 2245, 2018, pp. 336–345.

[44] P. Pourali and J. M. Atlee, "An empirical investigation to understand the difficulties and challenges of software modellers when using modelling tools," in *Proc. 21st ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2018, pp. 224–234, doi: 10.1145/3239372.3239400.

[45] H. Störrle, "Improving model usability and utility by layered diagrams," in *Proc. 10th Int. Workshop Modeling Softw. Eng.*, New York, NY, USA, May 2018, pp. 59–66, doi: 10.1145/3193954.3193958.

[46] H. Agt-Rickauer, R.-D. Kutsche, and H. Sack, "DoMoRe—A recommender system for domain modeling," in *Proc. 6th Int. Conf. Model-Driven Eng. Softw. Develop.*, 2018, pp. 71–82, doi: 10.5220/0006555700710082.

[47] A. Ribeiro, L. de Sousa, and A. R. D. Silva, "Comparative analysis of workbenches to support DSMLs: Discussion with non-trivial model-driven development needs," in *Proc. 4th Int. Conf. Model-Driven Eng. Softw. Develop.*, 2016, pp. 323–330, doi: 10.5220/0005745603230330.

[48] S. A. Safdar, M. Z. Iqbal, and M. U. Khan, "Empirical evaluation of UML modeling tools—A controlled experiment," in *Modelling Foundations and Applications*. Cham, Switzerland: Springer, 2015, pp. 33–44, doi: 10.1007/978-3-319-21151-0_3.

[49] Y. E. Ahmar, S. Gerard, C. Dumoulin, and X. Le Pallec, "Enhancing the communication value of UML models with graphical layers," in *Proc. ACM/IEEE 18th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS)*, Sep. 2015, pp. 64–69, doi: 10.1109/MODELS.2015.7338236.

[50] J. J. Ferreira, C. S. de Souza, and R. Cerqueira, "Characterizing the tool-notation-people triplet in software modeling tasks," in *Proc. Brazilian Symp. Hum. Factors Comput. Syst.*, 2014, pp. 31–40.

[51] J. Williams, N. Matragkas, D. Kolovos, S. Ananiadou, and R. Paige, "Software analytics for MDE communities," in *Proc. 1st Workshop Open Source Softw. Model Driven Eng.*, 2014, pp. 1–10.

[52] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, "Industrial adoption of model-driven engineering: Are the tools really the problem?" in *Proc. Int. Conf. Model Driven Eng. Lang. Syst.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 8107, 2013, pp. 1–17, doi: 10.1007/978-3-642-41533-3_1.

[53] N. Condori-Fernández, J. I. Panach, A. I. Baars, T. Vos, and Ó. Pastor, "An empirical approach for evaluating the usability of model-driven tools," *Sci. Comput. Program.*, vol. 78, no. 11, pp. 2245–2258, Nov. 2013, doi: 10.1016/j.scico.2012.07.017.

[54] A. Kuhn, G. C. Murphy, and C. A. Thompson, "An exploratory study of forces and frictions affecting large-scale model-driven development," in *Model Driven Engineering Languages and Systems*. Berlin, Germany: Springer, 2012, pp. 352–367, doi: 10.1007/978-3-642-33666-9_23.

[55] A. E. Kouhen, C. Dumoulin, S. Gerard, and P. Boulet, "Evaluation of modeling tools adaptation," HAL Open Sci., Paris, France, Tech. Rep. 00706701, Jun. 2012.

[56] H. Ranjna, "A comparative study of UML tools," in *Proc. Int. Conf. Adv. Comput. Artif. Intell.*, New York, NY, USA, 2011, pp. 1–4, doi: 10.1145/2007052.2007053.

[57] J. I. Panach, N. Condori-Fernández, A. Baars, T. Vos, and O. Pastor, "An experimental usability evaluation framework for model-driven tools," in *Proc. Congreso Interacción Persona-Ordenador*, 2011, pp. 67–76.

[58] H. Eichelberger, Y. Eldogan, and K. Schmid, "A comprehensive survey of UML compliance in current modelling tools," in *Software Engineering*. Bonn, Germany: Gesellschaft für Informatik e.V., Jan. 2009, pp. 39–50.

[59] J. de Sousa Saraiva and A. R. da Silva, "Evaluation of MDE tools from a metamodeling perspective," *J. Database Manage.*, vol. 19, no. 4, pp. 21–46, Oct. 2008, doi: 10.4018/jdm.2008100102.

[60] M. Auer, L. Meyer, and S. Biffl, "Explorative UML modeling—Comparing the usability of UML tools," in *Proc. 9th Int. Conf. Enterprise Inf. Syst.*, Jun. 2007, pp. 466–473.

[61] H. Störrle, "Large scale modeling efforts: A survey on challenges and best practices," in *Proc. 25th Conf. IASTED Int. Multi-Conf.*, Feb. 2007, pp. 382–389.

[62] A. Egyed, "Instant consistency checking for the UML," in *Proc. 28th Int. Conf. Softw. Eng.*, Shanghai, China, May 2006, p. 381, doi: 10.1145/1134285.1134339.

[63] A. Seffah and J. Rilling, "Investigating the relationship between usability and conceptual gaps for human–centric CASE tools," in *Proc. IEEE Symposia Hum.-Centric Comput. Lang. Environ.*, Feb. 2001, pp. 226–231, doi: 10.1109/HCC.2001.995263.

[64] Nielsen Norman Group. (Sep. 30, 2021). *Usability Inspection Method Summary: Article by Jakob Nielsen*. Accessed: Sep. 29, 2021. [Online]. Available: https://www.nngroup.com/articles/summary-of-usability-inspection-methods/

[65] J. R. Lewis and J. Sauro, "Usability and user experience: Design and evaluation," in *Handbook of Human Factors and Ergonomics*. Hoboken, NJ, USA: Wiley, 2021, pp. 972–1015, doi: 10.1002/9781119636113.ch38.

[66] E. Planas and J. Cabot, "How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and papyrus," *Comput. Standards Interfaces*, vol. 67, Jan. 2020, Art. no. 103363, doi: 10.1016/j.csi.2019.103363.

[67] Wikipedia. (Oct. 9, 2020). *List of Unified Modeling Language Tools*. Accessed: Dec. 8, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=List_of_Unified_Modeling_Language_tools&oldid=982628543

[68] Dec. 8, 2020. *Papyrus*. Accessed: Dec. 8, 2020. [Online]. Available: https://www.eclipse.org/papyrus/

[69] X. Zhou, Y. Jin, H. Zhang, S. Li, and X. Huang, "A map of threats to validity of systematic literature reviews in software engineering," in *Proc. 23rd Asia–Pacific Softw. Eng. Conf. (APSEC)*, 2016, pp. 153–160, doi: 10.1109/APSEC.2016.031.

[70] K. van Turnhout, A. Bennis, S. Craenmehr, R. Holwerda, M. Jacobs, R. Niels, L. Zaad, S. Hoppenbrouwers, D. Lenior, and R. Bakker, "Design patterns for mixed-method research in HCI," in *Proc. 8th Nordic Conf. Hum.-Comput. Interact., Fun, Fast, Foundational*, New York, NY, USA, Oct. 2014, pp. 361–370, doi: 10.1145/2639189.2639220.

[71] J. W. Creswell and C. N. Poth, *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. Thousand Oaks, CA, USA: SAGE Publications, 2016.

[72] U. Flick, E. von Kardoff, and I. Steinke, *A Companion to Qualitative Research*. Thousand Oaks, CA, USA: SAGE, 2004.

[73] P. Henry. (2015). *Rigor in Qualitative Research: Promoting Quality in Social Science Research*. Accessed: May 4, 2022. [Online]. Available: https://www.semanticscholar.org/paper/Rigor-in-Qualitative-research%3A-Promoting-quality-in-Henry/83c48b9bd79cafda8a877f87acb83ddc5c3babf6

REYHANEH KALANTARI received the B.S. degree in computer engineering from Semnan University, Iran, in 2012, and the M.S. degree in IT management from Tarbiat Modares University, Iran, in 2016. She is currently pursuing the Ph.D. degree in digital transformation and innovation with the University of Ottawa, Canada.

She worked as a Researcher and a System Administrator at two corporations in Iran, from 2016 to 2019. Since 2020, she has been working as a Research and Teaching Assistant at the University of Ottawa. Her current research interest includes user experience in software modeling tools.

TIMOTHY C. LETHBRIDGE (Senior Member, IEEE) received the B.C.S. and M.C.S. degrees from the University of New Brunswick, Canada, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from the University of Ottawa, Canada, in 1994.

He was a Member of Scientific Staff at Bell-Northern Research, Ottawa, ON, Canada, from 1987 to 1989, and has been a Faculty Member at the University of Ottawa, since 1994. His research interests include usable software engineering tools, software modeling, code generation, and software engineering education.

Dr. Lethbridge is a Senior Member of the ACM and a fellow of the Canadian Information Processing Society (CIPS). His awards include the IEEE Computer Society TCSE Outstanding Educator Award in 2016 and the IEEE Outstanding Contribution Award for his involvement as a Curriculum Co-Chair of the SE2004 Committee. He is a Professional Engineer (P.Eng.) and Information Systems Professional (I.S.P.).