JOT
JOURNAL OF
OBJECT TECHNOLOGY

Journal of Object Technology | **RESEARCH ARTICLE**

User Modeling in Model-Driven Engineering: A Systematic Literature Review

Aaron Conrardy[*,†], **Alfredo Capozucca**[†], **and Jordi Cabot**[*,†]
[*]Luxembourg Institute of Science and Technology, Belval, Luxembourg
[†]University of Luxembourg, Belval, Luxembourg

ABSTRACT In software applications, user models can be used to specify the profile of the typical users of the application, including personality traits, preferences, skills, etc. In theory, this would enable an adaptive application behavior that could lead to a better user experience. Nevertheless, user models do not seem to be part of standard modeling languages nor common in current model-driven engineering (MDE) approaches. In this paper, we conduct a systematic literature review to analyze existing proposals for user modeling in MDE and identify their limitations. The results showcase that there is a lack of a unified and complete user modeling perspective. Instead, we observe a lot of fragmented and partial proposals considering only simple user dimensions and with lack of proper tool support. This limits the implementation of richer user interfaces able to better support the user-specific needs. Therefore, we hope this analysis triggers a discussion on the importance of user models and their inclusion in MDE pipelines. Especially in a context where, thanks to the rise of AI techniques, personalization, based on a rich number of user dimensions, is becoming more and more of a possibility.

KEYWORDS User model, User modeling, Model-driven engineering, Low-code, Systematic literature review

1. Introduction

Model-Driven Engineering (MDE) (and its variations such as model-based software engineering, low-code software development, etc.) focuses on the usage of modeling for the development of complex software applications, with the purpose of increasing efficiency and effectiveness in software development (Brambilla et al. 2012).

In any of these complex applications, the human aspects play a key role (Grundy 2021). Indeed, if one talks about modeling a complete software application, modeling the user(s) of the application is also important. Generally, applications aim to help the users achieve their goals while providing them with the best possible user experience. For that purpose, both the users' behaviors but also their characteristics (preferences, cultural background, accessibility, etc.) need to be modeled, particularly the latter as reflected by the need for equality in today's software world (Robinson et al. 2020).

Additionally, emerging technologies, both hardware (e.g. IoT devices, VR/AR or generally more powerful components) and software (e.g. machine learning (ML) or large language models (LLMs)), allow for the creation of new types of applications, with a higher degree of personalization, which was not the case in the past (Planas et al. 2021). For example, a recent study has attempted to infer the fatigue level of the user by combining the data from a smartwatch and ML models (Liu et al. 2023), showing that new ways of profiling and adapting to users are emerging. Indeed, we see richer types of interactions and new possibilities for adaptation (e.g. tuning the response text of an AI-assistant to the personality or language skills of the user asking the question) in these new modalities of human-computer interaction.

Yet, there appears to be a lack of focus on modeling users in existing MDE approaches (Grundy 2021; Liebel et al. 2024; Abrahão et al. 2017; Michael et al. 2023), which hampers leveraging these new possibilities in MDE-driven development processes. To evaluate the state of user modeling in MDE, we

JOT reference format:
Aaron Conrardy, Alfredo Capozucca, and Jordi Cabot. *User Modeling in Model-Driven Engineering::A Systematic Literature Review*. Journal of Object Technology. Vol. 24, No. 2, 2025. http://dx.doi.org/10.5381/jot.2025.24.2.a12

performed a systematic literature review (SLR), collecting different attempts and methods to model users, the applications they were applied in and the provided tool support. Based on these results, we argue for the need of a renewed emphasis on user modeling and, in particular, the need for a unified user model, covering a rich number of dimensions, to facilitate the specification of complex user profiles. These models would then enable code-generation processes targeting the development of applications with adaptive features such as intelligent user interfaces.

The rest of the paper is structured as follows: in Section 2, we go over relevant definitions. Section 3 defines the need for an SLR. Section 4 describes the process of the SLR, providing the research questions, used query and libraries, inclusion/exclusion criteria, data extraction and data synthesis. The results are presented in Section 5 and discussed in Section 6. Section 7 describes the threats to validity of the performed review. Conclusions and next steps about how to use the SLR's results are given in Section 8.

2. Preliminary definitions

A *user model* describes one or more application users, also called end-users, including information on the user dimensions that are relevant to the application at hand (Rich 1979; Purificato et al. 2024). Specifically, dimensions intrinsic to the users, such as preferences, goals, behaviors or demographic details, are represented in a structured manner as part of a user model. In theory, these models should then let the application know its user(s) and provide services adapted to such user dimensions. Note that any application with the potential to adapt to the user is considered here. Thus, developers using a software development tool, for example, are also regarded as users.

In MDE, this definition remains largely valid (Abrahão et al. 2017). Only the importance of a concrete formalization increases. Indeed, a core element in MDE is the specification of models using a specific modeling language (or domain-specific language) formalized via a metamodel or a grammar. Both define the set of elements that could be part of a user model and how those elements can be combined among them.

The term *user modeling* refers to the creation process of a user model in an application (Purificato et al. 2024). Essentially, given a user and the different dimensions of the user (meta)model, user modeling involves assigning to a user a set of values for each dimension. This is also known as the *user profile*. One usually differentiates between static user dimensions that remain the same during a given runtime session (e.g. age), and dynamic ones, that could continuously be updated (e.g. mood).

3. Motivation

Proposing metamodels or grammars for different modeling perspectives is recurrent in the MDE community (Paige & Cabot 2024). While a call to establish a unified user model has been made in the past (Abrahão et al. 2017), the inclusion of human factors in MDE still seems to be an open problem (Liebel et al. 2024; Michael et al. 2023).

User models, user features, human factors have all been studied in past MDE research works but the domain lacks both a unified user model and methods on how incorporate it during the engineering process.

This need for a unified user model along with the MDE processes to exploit it is exacerbated by the advent of new AI techniques. Therefore, to determine how user models can be used not only to develop AI systems, but also how AI can contribute to define such user models, it is first required to know where we are standing.

This is exactly the goal of the present SLR. To the best of our knowledge, there is no systematic study that attempts to analyse the state of user models in MDE. The most relevant work we could find was a recent study from (Gaspar et al. 2024). They performed a targeted literature review to find dimensions of user models that can leveraged for intelligent user interfaces, only taking dimensions into account that can be learned through the user interaction. This led to them restricting the scope of their search results by focusing on a specific type of application and dimensions. Their results do not cover the other RQs targeted in this SLR, presented in the next section.

4. Research Methodology

This section describes the process followed to conduct the SLR, which mainly adhered to the well-established guidelines defined by Kitchenham (Kitchenham & Charters 2007). SLRs have the purpose to provide an overview of the existing contributions for a specific research question or topic, while guaranteeing an objective, reproducible and exhaustive result. The guidelines expect establishing first the need for the SLR, which was done in Section 3.

4.1. Research Questions

We defined four research questions (RQ) that focus on the aspects that help to grasp the state of the art of user modeling in MDE and identify gaps:

RQ1: Which dimensions of the user are modeled?

RQ1 explores the user dimensions that are represented in the existing user models. As mentioned in Section 2, these are the intrinsic dimensions of users. We excluded other more extrinsic dimensions (e.g. the user's current location or connected devices) which define more the context the users find themselves in more than the users' profiles. We acknowledge that extrinsic dimensions are also relevant for adaptive applications, yet, as these are not inherently part of the user, they do not fit in the user model. Instead, as seen in some studies (e.g., (Aarab et al. 2016; Motti & Vanderdonckt 2013)), user models are combined with device or environment models to build a context model, showing that these are separate concerns.

Beyond a pure list of dimensions, RQ1 also aims to establish the popularity of such dimensions, based on the number of sources that mention each dimension.

RQ2: How is the user model used?

RQ2 investigates the actual usage scenarios of user models in software applications. This includes the domain of the applications (e.g., education, health, etc.) for which the user model was created and for what purpose the user model was created (e.g., runtime adaptation, system risk analysis, etc.).

RQ3: Are the user dimensions fixed or dynamically evolving?

RQ3 aims to find out if current methods to populate user models are just static approaches where user values are set once at the beginning, or whether they support dynamic dimensions that are continuously adapted during the user interaction with the application. Obviously, dynamic approaches are more challenging to build but enable a more fine-grained and evolving response to the user's changing profile.

RQ4: How are user models implemented in a given application?

RQ4 looks at the extent of the tool support for user models. This RQ first looks at how the user model itself is formalized, i.e. are the dimensions just enumerated or formalized in some kind of grammar, (meta)model, or any other type of formalization? Then, it also assesses whether the user model can be automatically processed to generate some type of software component that would facilitate the personalization of the target applications.

This RQ helps to determine how advanced the level of integration of the user model in current MDE pipelines, and which technologies are more often employed to support such an integration and specification.

4.2. Search and Selection Strategy

Figure 1 depicts the search and selection process.

4.2.1. Search String As our goal was to find user model proposals in MDE approaches, we divided our search query into two fragments that narrow the search to the MDE field and user models respectively. The fragments consist of individual terms that were combined using the boolean operators OR and AND as follows:

("model-driven" OR "model-based-software-engineering" OR "MDE" OR "MBSE" OR "MDA" OR "low-code" OR "no-code" OR "metamodel" OR "meta-model" OR "domain-specific-language" OR "DSL" OR "grammar")
AND
("user/human model" OR "user/human profile" OR "user/human characteristic" OR "user/human dimension" OR "user/human attribute" OR "user/human factor" OR "user/human ontology" OR "model the end-user/user/human" OR "model end-user/user/human/person/people" OR "model(l)ing end-user/user/human/person/people")

The search query has gone through numerous updates to improve the results and the number of results, the latter in an attempt to neither limit nor explode the number of results. We limited the query to look up words in the keywords, titles and abstracts. Additionally, we limit the results to journal, conferences and short papers written in English.

The MDE fragment contains the pre-fix "model-driven", the most common acronyms used in the domain, the core concepts "metamodel", "domain-specific-language" and "grammar" (Brambilla et al. 2012) and the terms "low-code/no-code" as these are considered synonymous or variations of MDE (Di Ruscio et al. 2022; Tosi et al. 2024; Cabot 2024). We omitted the words "modeling" and "ontology" as stand-alone terms, as these led to numerous irrelevant results. The user model fragment consists of terms relating to the modeled characteristics ("human/user" combined with words such as "characteristics", "attributes", etc.), the produced artifacts ("human/user" and "model/ontology") and the action of modeling the users (e.g. "model user" or "model(l)ing user"). For the latter, we covered both the US and UK English spellings by including "modeling" and "modelling" in the query.

4.2.2. Digital Libraries We decided to include the following digital libraries for conducting the search, as these are regarded as the most relevant within the software engineering domain (Kitchenham & Charters 2007; Brereton et al. 2007):

- ACM Digital Library (ACM): https://dl.acm.org
- IEEEXplore (IEEE): https://ieeexplore.ieee.org
- SpringerLink (SL): https://link.springer.com
- ScienceDirect (SD): https://www.sciencedirect.com
- Scopus: https://www.scopus.com

Note that, for each library, the available search engine followed a different syntax. Thus, the defined query had to be adapted for each library. Furthermore, as neither SpringerLink nor ScienceDirect support a search based on title and abstract, we completed a full-text search. Scripts were then developed to perform the filtering at title and abstract level, which are part of the replication package for this SLR (Conrardy 2024). The initial search was conducted on September 12, 2024, and resulted in 526 unique (578 with duplicates) papers.

4.2.3. Selection Strategy Once the initial set of unique papers was collected, we proceeded with the screening at the title/abstract level and then at the full-text level. The screening was done following a set of exclusion and inclusion criteria:

- **Exclusion**:
  - Uses terms from the query in a different way than what we intended (e.g., 3D human model, model of human cells or a natural language's grammar).
  - Not a primary study.
- **Inclusion**:
  - Proposes an explicit user model proposal formalized as a metamodel, grammar, JSON schema, etc.

During the title/abstract screening, if it was unclear whether the exclusion criteria could be applied, the paper was kept for a full-text screening. Additionally, if multiple papers from the same authors presented the same (or an updated) solution, the most recent version was kept. In the title/abstract screening, 228 papers were kept (298 removed) and in the full-text, 22 papers were kept (206 removed). Forward and backward snowballing (Wohlin 2014) was performed on the kept papers, increasing the number of total papers to 30. For a part of the selection process, the CADIMA[1] tool was used (Kohl et al. 2018). Specifically, CADIMA was used for the following tasks:

[1] https://www.cadima.info/

1. Automatically recognize duplicates from a given list of references.
2. Propose an interface to perform the title/abstract and full-text screening.
3. Generate an Excel file containing post-screening results.

4.3. Data Extraction and Synthesis

To organize the collected data, an Excel template was created. The filled Excel file is available in our replication package (Conrardy 2024). For each paper, we extracted the title, the authors' names, the DOI, the venue, the year of publication and the number of citations excluding self-citations. Additionally, we manually added a short summary with the main takeaways. This summary played a key role in the next steps of the SLR as it reduced the times we had to come back to the paper for full inspection of its content.

Regarding the data gathering to answer the research questions, for RQ1, we started with a set of initial dimension categories inspired from previous works such as (Abrahão et al. 2021; Purificato et al. 2024). Next, for every paper, we would then extract the dimensions proposed in that paper and map them to the existing columns, adding new ones if necessary. For RQ2, we iteratively extended the columns based on new domains or types of applications we found in the papers. For both RQ1 and RQ2, we took the liberty to generalize some of the extracted data, to obtain a simplified categorization. For example, if a reviewed paper proposes a user model that accounts for hearing at different frequencies, we would map it simply to the dimension 'Hearing,' omitting granular details. Likewise, if a user model is used to adapt a conversational agent, we categorize it under 'Content Adaptation'. For RQ3 and RQ4, the columns needed for the data extraction were set from the beginning and did not need any updates.

As mentioned earlier, if the same solution appeared in multiple papers authored by the same people, the most recent one was kept. In an effort to be thorough, we still revisited all available papers related to the same solution to avoid missing any relevant information.

5. Results

This section presents the insights gained during the analysis of the collected data. In the end, out of the initial 526 unique papers 30 papers went through a rigorous review process. Table 1 contains the list of fully reviewed papers, the first authors, their publication year and the venue they were published in. We observe that the venues are not limited to ones focusing on MDE or software engineering in general. Indeed, while 10 paper were published in software engineering venues (ASE, HCSE, MODELSWARD, SoSym, MEDI, MODELS, CSER, ICWE), information system venues were also a popular choice with five papers published (JDM, EMCIS, RCIS, UAIS).

Figure 2 contains a histogram displaying the trend of publication over the years, where we recognize a slight decline in the number of papers. Additionally, we created a graph that showcases the citations between the selected papers in Figure 3. Out of the 30 papers we analysed, only eight of them appeared in the citations of five other papers from the chosen selection. Apparently, new proposals do not seem to deeply acknowledge and compare with previous ones at the risk of repeating themselves.

ID	Author-Year	Venue
P1	(Jaskolka & Hamid 2023)	ASE
P2	(El Hog et al. 2012)	ACM SAC
P3	(Brambilla & Tziviskou 2008)	ICWE
P4	(Aarab et al. 2016)	IEEE ACS
P5	(Wischenbart et al. 2012)	WWW
P6	(Jovanovic et al. 2014)	HCSE
P7	(Kennedy et al. 2020)	SysCon
P8	(Vázquez-Ingelmo et al. 2020)	MDPI
P9	(Khider. et al. 2020)	MODELSWARD
P10	(Garrigós et al. 2012)	JDM
P11	(Silva & Belo 2023)	EMCIS
P12	(Yigitbas et al. 2020)	SoSyM
P13	(Karam et al. 2012)	MEDI
P14	(Gaspar et al. 2024)	Multimed. Tools Appl.
P15	(Abbar et al. 2010)	iiWAS
P16	(Khalajzadeh et al. 2022)	MODELS
P17	(Nunes et al. 2013)	Interacting with Computers
P18	(Elmagrouni et al. 2016)	ICMCS
P19	(Perrotin et al. 2020)	MODELS
P20	(Bahaei & Gallina 2021)	ESREL
P21	(Ben Cheikh et al. 2012)	RCIS
P22	(Kaklanis et al. 2016)	UAIS
P23	(Orellana & Madni 2014)	CSER
P24	(Jaouadi et al. 2018)	SoSyM
P25	(Motti & Vanderdonckt 2013)	RCIS
P26	(Tapucu et al. 2008)	M-PREF
P27	(Bocanegra et al. 2015)	CCC
P28	(White et al. 2010)	CL
P29	(Park et al. 2018)	IEEE TPAMI
P30	(Ade-Ibijola 2017)	RobMech

Table 1 Table of selected papers

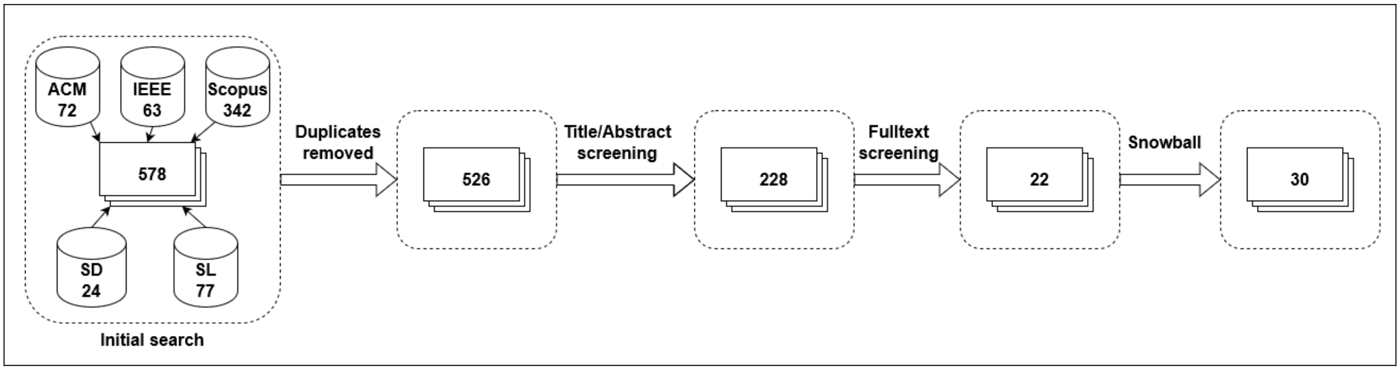

Figure 1 Search Process

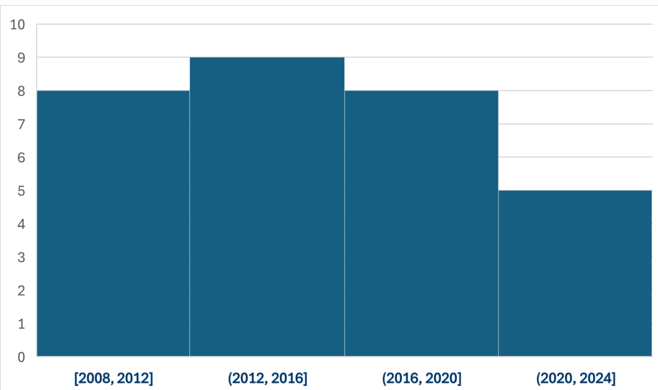

Figure 2 Histogram of publication year

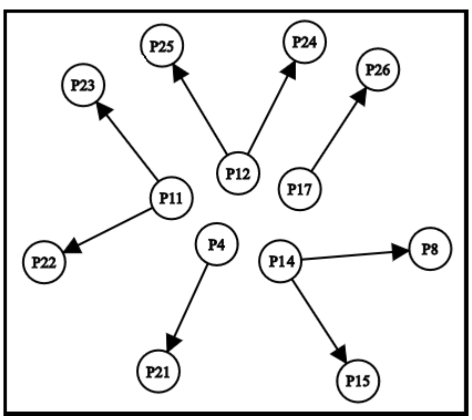

Figure 3 Paper citation graph (edge direction symbolizes citation direction)

5.1. RQ1: Which dimensions of the user are modeled?

The data extraction process provided a long list of dimensions to characterize a user. To better structure the results, we group the dimensions in a number of more generic categories:

- **User Competencies** describe how competent a user is given a task or in general by taking into account their knowledge, learned skills and other attributes reflecting their competency.
- **Personality** reflect the user's personality traits, that is, stable internal characteristics of the users.
- **Preferences** describe personal preferences regarding how an application does something, including the way the application interacts with the user, shows the user content, how the content is presented and more.
- **Demographic Information** tackle generic information about users that are application independent and usually static such as the name, age, nationality and gender of the user.
- **Accessibility** features describe accessibility needs of the user or any attribute that might affect these.
- **Emotions & Mood** describe the different emotions or moods the users might experience while interacting with the application.
- **Goals** describe any kind of goal (related to an application or not) users might have.
- **Generic property** is used if in a user (meta)model, no concrete attribute is specified and a generic pair of <dimension,value> like solution is proposed.

Beyond the categories inspired from (Abrahão et al. 2021; Purificato et al. 2024), we added the accessibility category.

Note that, dimensions mapped to one category have the potential to affect other categories or dimensions. An example would be the possibility that the dimension "Age" could definitely affect dimensions in the "Accessibility" or "User Competencies" categories.

Table 2 shows the list of categories and dimensions and the papers (and total number) that included such dimensions in their proposal. While some dimensions, such as age or nationality, are self-explanatory, we provide a short description for the less obvious ones in Table 3 based on the information from the reviewed papers and our own interpretation. A high level view of the coverage of the dimensions categories can be seen in Figure 4 and Figure 5 contains the number of dimensions per paper.

At first glance, we notice that the most popular categories of user dimensions would be the ones describing the users' competences (50.0%), preferences (50.0%) and demographic information (63.3%). With a smaller yet still relevant popularity, accessibility (33.3%) dimensions seemed to still be fairly common when modeling users. Finally, the least common dimen-

Dimension Category	Dimension	Paper	#Papers
User Competencies	Role/Job	P1, P4, P5, P7, P9, P10, P12, P13, P14, P19, P23	11
	Knowledge/Expertise	P1, P4, P7, P8, P11, P13, P14, P19, P23	9
	Skills/Skillset	P4, P7, P8, P9, P12, P14, P16, P19, P23	9
	Education	P5, P9, P14	3
	Experience	P7, P9, P11, P14, P19, P20	6
	Physical Capability	P20	1
	Mental Capability	P19, P20	2
	Reliability	P19	1
Personality	Attitudes	P1, P7, P11, P20	4
	Emotional Resistance	P19	1
	Robustness	P19	1
	Motivation	P11	1
	Bias	P8	1
Preferences	Generic Preference	P4, P6, P8, P10, P15, P16, P17, P27, P28	9
	Design/Presentation	P2, P18, P21, P22	4
	Interaction Modality	P2, P22	2
	Content	P3, P10	2
	Language	P12, P18	2
Demographic Information	Name	P2, P3, P4, P5, P9, P10, P12 P13, P16, P18, P19, P21, P22, P30	14
	Address	P4, P5, P14, P30	4
	Gender	P4, P5, P11, P12, P14, P16, P22, P24, P29, P30	10
	Relationships	P3, P5, P13, P16, P19, P20, P30	7
	Nationality	P14, P18	2
	Known Languages	P14, P16, P22	3
	Hobbies	P14	1
	Interests	P14	1
	Sexuality	P30	1
	Age	P4, P11, P12, P14, P18, P27, P29	7
Accessibility	Disability	P6, P7, P14, P21, P22, P23	6
	Sight	P6, P11, P12, P22, P29	5
	Hearing	P11, P22	2
	Motoric	P6, P11	2
	Cognitive	P6, P22	2
	Memory	P6, P20, P22	3
	Attention	P6, P20	2
	Sensory	P6	1
	Speech	P22	1
	Mobility	P11, P22	2
	Physical State	P20	1
Emotions & Mood	Emotion	P12, P14	2
	Mood	P12, P14, P25	3
	Stress	P11	1
	Fatigue	P11	1
Goals		P8, P14, P19	3
Generic Property		P8, P12, P16, P21, P22, P25, P26, P27	8

Table 2 Modeled dimensions in user model

sions were personality (20.0%), emotions and mood (16.7%), and goals (10.0%).

Dimension	Explanation
Knowledge	User's awareness, familiarity, and understanding of a topic.
Skills	Learned abilities to perform tasks effectively.
Physical Capability	The ability to perform physical actions, including endurance, coordination, and motor skills, required by given tasks.
Mental Capability	The capacity to process information, maintain attention, and make decisions under varying cognitive demands, required by given tasks.
Reliability	Consistency and dependability in performing tasks.
Attitudes	User's predisposition or stance towards certain topics or tasks.
Emotional Resistance	Ability to handle stress and emotional challenges.
Robustness	Stability in behavior and reactions under varying conditions.
Bias	Similar to "Attitude", yet based on faulty assumptions, leading to systematic deviations in judgment or unfair treatment of a target.
Content	Preferences for specific types of information or material.
Interaction Modality	Preferred way of interacting with an application (e.g., voice, touch, text).
Interests	Topics the user actively engages with through activities.
Hobbies	A specific interest pursued with recurrent and dedicated engagement.
Disability	Any physical or cognitive impairment affecting interaction.
Motoric	Describes components related to the user's ability to perform motor functions and fine movements.
Cognitive	Describes mental processes like perception and reasoning.
Sensory	User's ability to detect and respond to stimuli through sensory organs.
Mobility	Information on the user's ability to move and related components.

Table 3 User dimensions and their explanations

There were eight papers (26.7%) that showcased a user model that used a generic property to describe any user dimension. These models usually consisted of a property class that could be named and customized freely, thus the user model having the potential to include any kind of dimension.

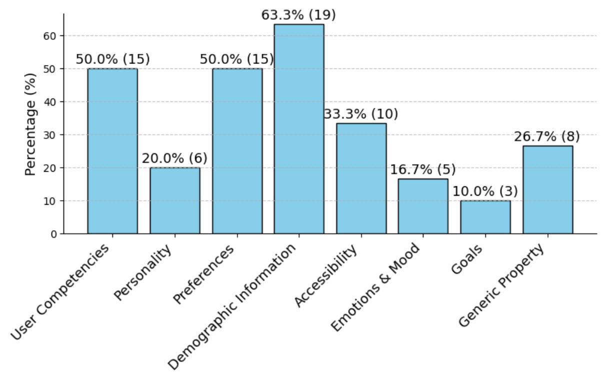

Figure 4 Number of papers covering each dimension category among the 30 reviewed

Excluding the generic category, the median of the number of categories covered by a specific proposal is two, with the maximum number of categories being five (for papers P11, P12, P14). Regarding the dimensions themselves, the median of dimensions contained in a user model is five, with the most being 14 (P14).

5.2. RQ2: How is the user model used?

Figures 6 and 7 depict the results of the mapping of each paper proposal to a specific purpose and to a concrete domain of application respectively.

Regarding the purpose of specifying a user model in a given work, the most popular reason was to enable adaptiveness in user interfaces (19/30). Especially the adaptation of the content itself (P2, P3, P4, P8, P9, P10, P12, P14, P15, P16, P17, P18, P21, P22, P24, P25, P26, P27, P28) or the way that the content is presented (P2, P6, P8, P12, P14, P16, P18, P22, P24, P25, P27) were the most recurring types of adaptation. An example would be adapting which content is shown to the user, or the language of such content. Less recurrent was the adaptation of the used modality to present content to the user (P2, P12, P16, P22, P24, P25).

Another popular purpose was modeling explicitly users to facilitate the interoperability and evolution of the system considering its users. In this context, user information is part of the data modeling efforts of the system (7/30) similar to the modeling of other system data. As an example, P7 adds a user model component to a model of organisational management systems, with the goal of providing a clearer overview of the participants in the system with a focus on the organization evolution.

Less commonly, user models are also used to tackle security and privacy aspects of software systems (5/30) or support software testing (1/30). For the former, some papers include the user in the model to perform risk analysis calculations to spot vulnerabilities (P19, P20, P22). P1 proposes the usage of the user model to support the development of secure software by recommending the implementation of security measures depending on the developer's competencies. In P30, the user models are used to generate synthetic profiles to avoid privacy issues when processing data. The only work tackling software testing,

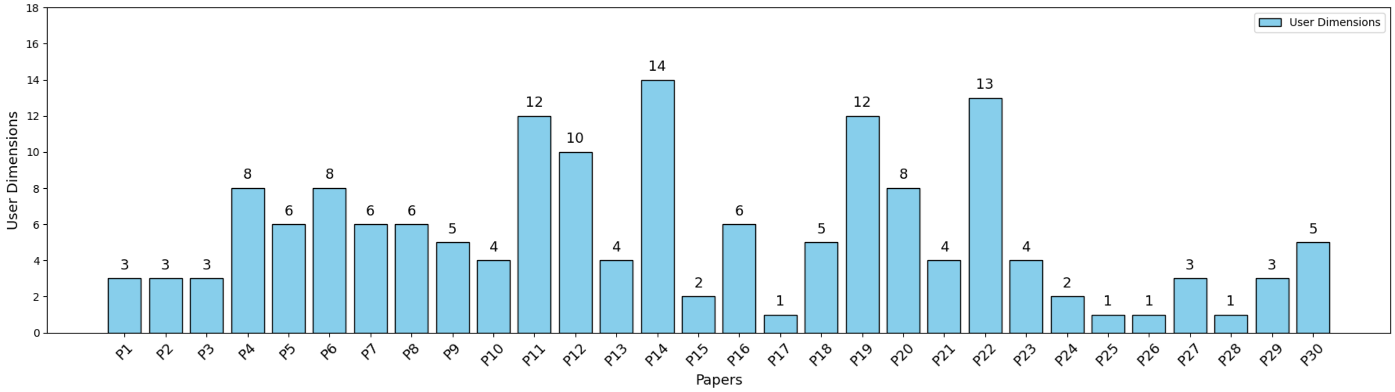

Figure 5 Number of user dimensions modeled in each reviewed paper

P6, proposes the idea to combine user characteristics defined in the model with usability tests to evaluate the usability of user interfaces.

The domain of application is very diverse. In fact, the largest share of proposed applications (10/30) are not tied to any specific domain.

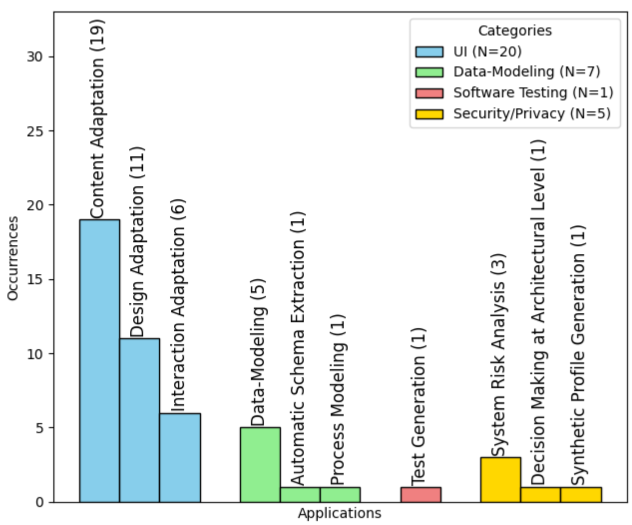

Figure 6 Categorization of application types by specific use cases of the user model across reviewed papers

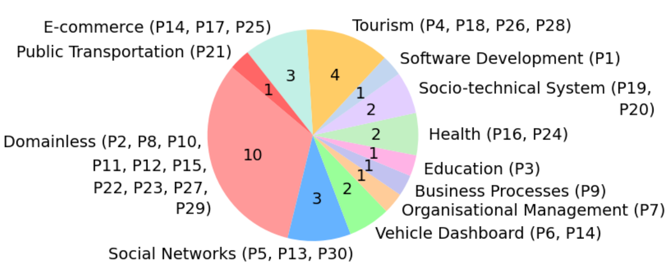

Figure 7 Distribution of domain of application

5.3. RQ3: Are the user dimensions fixed or dynamically evolving?

Table 4 answers this question by listing each paper in one of the two categories.

The classification was done based on a thorough reading of the proposal where we looked for descriptions of methods on how to populate or just measure a specific user dimension, regardless if such method was implemented or not. Additionally, the table contains a "Unclear/Not mentioned" column that reflects cases in which it was not clear how the dimensions were to be profiled. For example, P11 proposed various user dimensions, some we would even expect to be dynamic. Yet, no explanation is provided on how the application will profile the user, thus P11 being mapped to the "Unclear/Not mentioned" column.

As an example of each category, we can mention P1 in the static one, as P1 requires users to answer a predefined questionnaire to profile them. Once this step is completed, the dimensions of the user model are not expected to change again in the short term. P12 would be an example of the dynamic category for its mood dimension, periodically updated by capturing pictures of the user face and updating the mood value based on the analysis of the picture.

Overall, 24 models out of the 30 propose static dimensions, with four of these 24 also proposing dynamic dimensions as well. No model proposed only dynamic dimensions and for six papers it was not clear when or how the dimensions were supposed to be profiled.

Type of profiling	Papers
Static	P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P12, P13, P14, P15, P17, P19, P21, P22, P24, P26, P27, P28, P29, P30
Dynamic	P3, P4, P12, P29
Unclear/Not mentioned	P11, P16, P18, P20, P23, P25

Table 4 Type of profiling for user dimensions

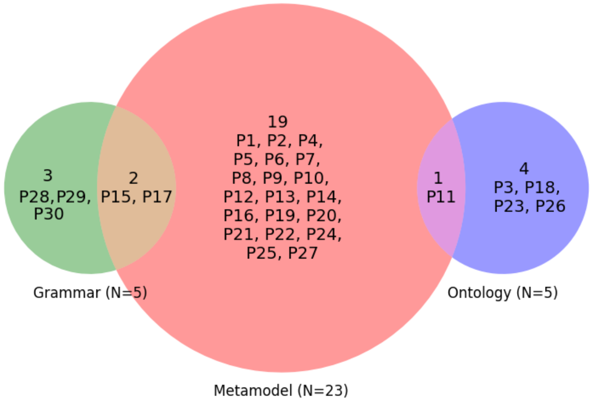

Figure 8 Venn diagram of employed formalization methods

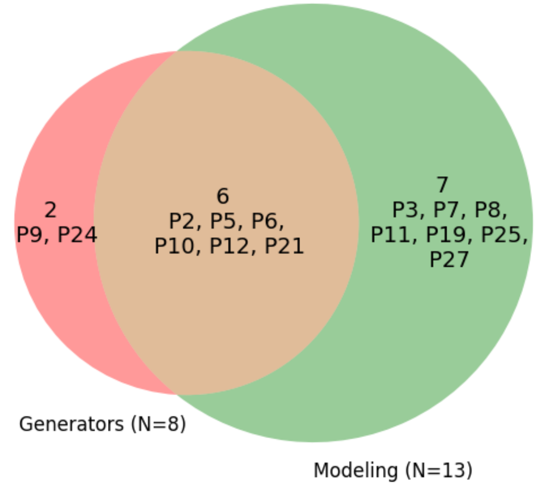

Figure 9 Available generators and modeling tools

5.4. RQ4: How was the user model implemented in a given work?

Figure 8 summarizes the techniques to formalize the user models. The most popular is metamodeling (23/30), typically described using UML class diagrams. There were five papers that described a user model using an ontology specified with the OWL language and five papers that defined a grammar to formalize their user model.

Moreover, in total, 13 papers (P2, P6, P7, P12, P13, P15, P19, P21, P25, P26, P28, P29, P30) actually developed the applications shown in RQ2 that somehow process or leverage the user models. Beyond running applications, tools supporting MDE-driven development processes were proposed as well. There are two main categories: modeling tools to enable the actual specification of users according to the user (meta)model proposed and generators able to transform these models into other software artifacts. This is shown in Figure 9.

More specifically, the eight papers that proposed some form of generator based on model transformation (model-to-model or model-to-text) targeted adaptive frontend code (P2, P6, P10, P12, P21, P24), the creation of user schemas to store user data (P5) or recommender systems (P9).

The generators' transformations were implemented using the technologies ATL (4/8), Xtend (2/8), XMI (1/8) and QVT (1/8).

Regarding the modeling tools, Eclipse[2] is the most used as base framework (5/13), due to the easiness of creating and adding plugins to their modeling platform (Yigitbas et al. 2020; Jaouadi et al. 2018). Only six papers provide both a modeling tool and generators that consume the user models (P2, P5, P6, P10, P12, P21).

In terms of numbers, 43% of the reviewed papers propose an actual application that integrates the user model, 27% provide generators that attempt to speed up the implementation of applications incorporating the user model and 43% provide a modeling environment.

[2] https://www.eclipse.org/

6. Discussion

This section contains more in depth interpretations and reflections on the results.

6.1. Missing dimensions in user models

As showcased in Section 5.1, the available user models already cover a wide range of user dimensions. Yet, we notice a trend that "simple" dimensions are modeled more often than "complex" ones. By "simple" we mean static dimensions that are easy to populate. Information such as the name, the language preference or the roles of the users tend to remain static and only require a user to enter them manually and their correctness is guaranteed, while features relating to personality, accessibility, or emotions change more frequently and are generally harder to profile. Obviously, simple dimensions are less useful when it comes to generate powerful application adaptations. This is evident when looking at the results of RQ3, as most works describe how to work with the static data, but do not tackle the dynamic aspects even though some of the proposed dimensions could be considered dynamic (P11 models "Fatigue", yet does not mention how to profile it).

And while current approaches propose already a good number of dimensions, there are quite a few still missing if we look outside MDE. Indeed, user modeling is a generally a wide research topic in computer science but also in other scientific domains such as sociology or psychology. The latter tends to focus on aspects related to the mental model or personality of humans, such as the big five models of personality (Babcock & Wilson 2020). While some of proposals do try to cover personality traits, they are rather limited, covering only one or two personality related dimensions. They could be significantly enriched by looking at how users are profiled in other domains. Similarly, user models could adopt existing user taxonomies focusing on cultural attributes (Plocher et al. 2021) or a wider range of emotion related attributes (Heckmann et al. 2005).

Therefore, we can argue that user models are biased towards simple dimensions and that a stronger focus on dynamic dimensions, enriched with dimensions coming from other scientific fields, is needed to get a more complete user model.

Some proposals try to use generic properties as way to allow for this extensibility but, in our opinion, this is a pragmatic but dangerous solution as the full application logic will depend on the interpretation of specific strings in the values of the generic property.

6.2. A fragmented and small community

We believe the importance of the topic does not really correspond to the number and impact of user modeling papers in the broad MDE community.

In part, we believe this is because user modeling community is very fragmented, with proposals not citing and building on top of previous approaches. See Figure 3 for an analysis of the citations among the papers in this domain. As a consequence, most papers that tackled similar topics or applications tended to create a concrete user model from scratch, thus wasting resources on re-inventing the wheel as most of the time, the final user model consists of elements present in older user models.

The lack of a unified model that could be used as core reference could also justify the small overall citation count of user modeling papers (see Figure 10). Note that for the citation count, we removed the self-citations. Even OMG[3] standards such as UML[4] or IFML[5], or other standards such as UsiXML[6] do not derive or use user models from the existing literature and only provide very limited expressiveness for the definition of user models, mostly restricted to access control scenarios. The fact that in some proposals the focus is not on the user model itself but, on the contrary, the user model is a secondary result also limits the visibility of the proposals.

We believe this situation highlights the need of a unified user model. If accepted by the main actors in this community, such user model could then be pushed as standard de facto and be adopted by other MDE approaches interested in reusing an existing user model instead of taking the time to develop their own. It could potentially even make its way up to standard languages.

Otherwise, we are also giving the message that taking into account the users' needs is not a priority in the applications we build, which we believe it is not the case.

6.3. Need for quality evaluation of user models

While most of the analysed papers specifically mention following MDE principles and use MDE terms, only one of them (P21 that checks the consistency between the user model and the application model) performs some kind of quality evaluation of the created user models, with three more mentioning model verification as future work, without getting into details.

Given the increasing importance of user models, we believe several of the model-based testing, verification and validation approaches could be applied to user models. For instance, we could check the intra-consistency of the model (e.g. a user that is hearing impaired is unlikely to declare audio as preferred communication modality or a user born in a certain country is likely to have some mastery of the local language(s)). At the inter-consistency level we could put in place some rules to guarantee, for instance, that user dimensions are part of the adaptations declared in the UI. On the model verification front we could make sure the user model is satisfiable, i.e. we can actually instantiate it in a way that all consistency and well-formedness rules evaluate to true (e.g. age cannot be over 200, a person cannot be in two contradictory moods at the same time, etc.).

6.4. Interplay between machine learning and user models

As discussed in RQ3, dynamic approaches are the exception. In part because they are more difficult to manage as they require an automatic process able to infer the values for the dynamic dimensions on a recurrent basis.

We believe ML could be a key factor in implementing such dynamic approaches and, therefore, increasing the number of proposals focusing on dynamic dimensions (e.g. many of the approaches were created at a time where technology was not as advanced as today (Abrahão et al. 2021)). Technologies like Natural Language Processing (NLP) or image recognition could be used to populate several dimensions with a high degree of reliability. For instance, NLP can be used to assess the language skills of a user when interacting with the application via a chatbot, and thus, enable the adaptation of the chatbot response to a vocabulary and grammar suitable for that skill. As also seen before, image recognition could for example be used to infer the mood of the user. Only three papers (P8, P14, P16) mention as future work to explore the usage of ML to make predictions about user preferences or classify users based on their behavior and only two papers (P12, P29) actually used ML to profile the user. Yet, these were more recent publications, which could indicate that the popularity and increased accessibility of ML tools will lead to an increase in using them when modeling users.

The other direction (user models as input of ML pipelines to produce AI components that are better tailored to the user profiles) is also a promising approach to increase the awareness and impact of user models. We have not yet seen an explicit use of user models in the ML field which is surprising given that a significant portion of machine learning tasks leverage user data (Purificato et al. 2024) and a significant amount of MDE works already tries to get closer to the needs of machine learning (Naveed et al. 2024). We believe there is an opportunity to leverage MDE to enhance ML models and algorithms with user models, by providing easy-to-use pipelines to train ML components with user data, for example, for classification purposes.

6.5. Limited exploitation of the user models

As revealed in RQ4, few papers come with tool support to exploit the user models. As such the return on investment for modeling user models is very low as we cannot leverage to automate other parts of the application.

[3] https://www.omg.org
[4] https://www.uml.org/
[5] https://www.ifml.org/
[6] http://www.usixml.org

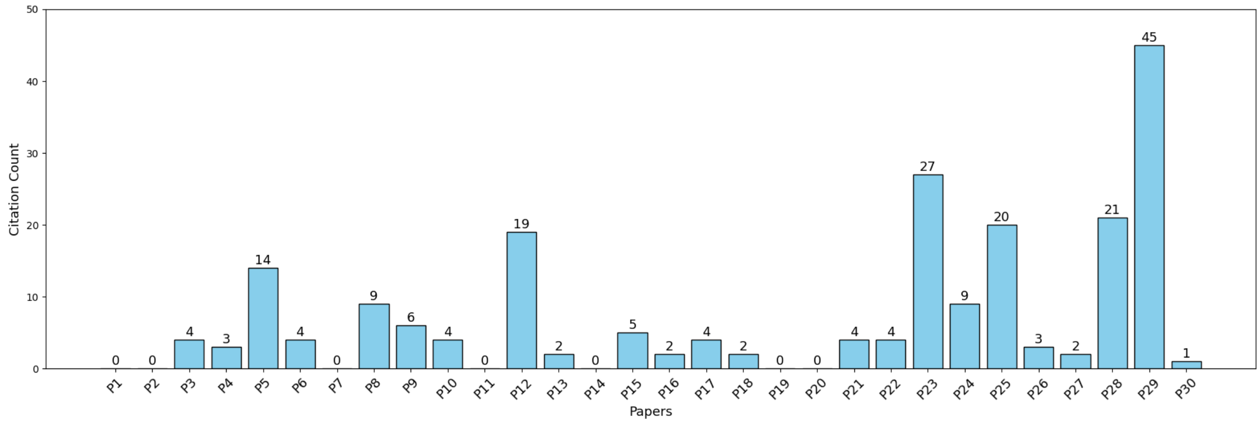

Figure 10 Citation count per paper and the amount of covered user dimensions

At the very least, we believe each user model proposal should come with the modeling editor but also with a number of generators or interpreters able to parse the user model and produce as output some software artefact that enhances the underlying software application.

We are fully aware that research projects experience the valley of death phenomena (DiMario & Hodges 2023) by which artifacts developed for research projects rarely make it to the industry and rather disappear once the project ends. Indeed, investing time to create proper tool support for user models is challenging. We hope that if we first agree to try to combine better the different approaches into a unified solution we could then also joint forces to create a suite of user modeling components that could be reused and expanded later on in new projects willing to exploit user information for a variety of domains and applications.

Eclipse could still be the basis for that, e.g. to facilitate the interoperability among the solutions, but other alternatives are possible, e.g. based on Graphical Language Server Platforms (Metin & Bork 2023).

7. Threats to validity

We will now briefly discuss aspects that may affect the validity of our results following the validity constructs defined by Wohlin et al. (Wohlin et al. 2012). Note that we omit external validity, due to the nature of SLRs to possess a fixed scope, thus limiting the potential for generalization by its own nature.

7.1. Construct validity

Construct validity refers to the relevance of the chosen primary sources to answer the RQs. We reduced this threat by choosing relevant libraries for the domain of software engineering, as mentioned in Section 4.2.2. Additionally, the inclusion and exclusion criteria were clearly defined and discussed by all the authors. The same goes for the search query, that contains the synonyms or terms we felt are relevant to find relevant works. Arguably, some terms are missing from the query that could have further guaranteed finding all of the relevant results. Most notably, additional terms that could describe a user are missing, such as developer, patient or citizen to name a few. Additionally, by imposing the keywords focusing on MDE and omitting terms such as "adaptive" or "context", which are often linked to user models, there is the risk to miss out on results that implicitly contribute to MDE. Although we may miss some potential correlations, we mitigated this risk by applying forward and backward snowballing, effectively increasing result coverage.

7.2. Conclusion validity

Threats to conclusion validity are concerned with how reliably we can draw conclusions about the wished or expected goal and the outcomes of the study. In our case, the systematic process of the SLR that follows the Kitchenham guidelines already reduces this threat. It is further reduced by the definition of the data extraction form based on the RQs. Multiple refinements took place after joint discussions on the RQs and the data form to assure its quality and to make sure that data relevant to the RQs was extracted.

7.3. Internal validity

While some phases of the data extraction were developed by single reviewer, methods to ensure objective results were implemented to avoid a selection bias:

- During the selection process, random included and excluded samples were selected by the authors to verify whether the inclusion/exclusion criteria were valid and applied correctly.
- A replication package is available to re-create the steps of the SLR.
- Regular meetings between all authors took place to discuss the validity of the following aspects of the SLR:
  - The search query.
  - The choice of digital libraries.
  - The conclusions reached based on the extracted data.
- The systematic nature of the SLR following the well-established guidelines (Kitchenham & Charters 2007) is objective in itself, such as the data extraction based on pre-defined technical questions.

8. Conclusion and roadmap

In this paper, we conducted an SLR on the state of user modeling in the MDE domain. Results show a diverse set of disconnected proposals, covering a partial number of dimensions with an emphasis on those characteristics that are easier to profile. Moreover, most dimensions are regarded as fixed instead of allowing their dynamic evolution during the interaction with the software application. It is also worth noting that tool support is also rather limited, mostly limited to enabling the creation of the user models itself.

The roadmap we hope to see in this area stems from the discussion points seen above. Most importantly, we believe the community should agree on a unified and re-usable user model. From the reviewed works and our own interpretation, we identified challenges that hinder the development of or could explain the lack of a unified user model. First and foremost, as seen by the different dimension categories presented in this work, modeling a human is a multidisciplinary endeavor, requiring knowledge in fields such as psychology or medicine. Yet, experts are not always present during the creation of models, leading to uncertainties and a lack of completeness when creating such models. Secondly, a choice has to be made to avoid modeling too much or not enough, as something as knowing the user's finger nail length does not have much value, while missing out on the user's preferred interaction language would lead to a decreased user experience. Essentially, knowledge on which dimensions are relevant to increase the user experience in a personalized application is necessary.

A first step toward a unified user model results from this SLR, as the collected dimensions could be used a basis to create such a model, enriched with dimensions learned from user profiling in other domains (e.g., sociology). The first challenge would partially be covered, as some of the reviewed works were created with domain experts. Nonetheless, once a first unified user model is developed, experts should be contacted as to review and improve the model. These could provide feedback if whether the dimensions are represented correctly, but also whether something is missing that could further increase the possible personalization opportunities.

On the technical side, we expect to see a new generation of ML-based proposals to automatically and incrementally derive a user profile from the analysis of user interactions and a number of automatic pipelines able to transform the user information in concrete application adaptations that personalize the application to cater to the user's needs and profile.

Acknowledgments

This project is supported by the Luxembourg National Research Fund (FNR) PEARL program, grant agreement 16544475.

References

Aarab, Z., Saidi, R., & Rahmani, M. D. (2016). Event-driven modeling for context-aware information systems. In *2016 ieee/acs 13th international conference of computer systems and applications (aiccsa)* (p. 1-8). doi: 10.1109/AICCSA.2016.7945785

Abbar, S., Bouzeghoub, M., & Lopes, S. (2010). Introducing contexts into personalized web applications. In *Proceedings of the 12th international conference on information integration and web-based applications & services* (p. 155–162). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1967486.1967513

Abrahão, S. M., Insfrán, E., Sluÿters, A., & Vanderdonckt, J. (2021). Model-based intelligent user interface adaptation: challenges and future directions. *Software and Systems Modeling*, *20*, 1335 - 1349. doi: 10.1007/s10270-021-00909-7

Abrahão, S., Bordeleau, F., Cheng, B., Kokaly, S., Paige, R., Stöerrle, H., & Whittle, J. (2017). User experience for model-driven engineering: Challenges and future directions. In *2017 acm/ieee 20th international conference on model driven engineering languages and systems (models)* (p. 229-236). doi: 10.1109/MODELS.2017.5

Ade-Ibijola, A. (2017). Synthesis of social media profiles using a probabilistic context-free grammar. In *2017 pattern recognition association of south africa and robotics and mechatronics (prasa-robmech)* (p. 104-109). doi: 10.1109/RoboMech.2017.8261131

Babcock, S. E., & Wilson, C. A. (2020). Big five model of personality. In *The wiley encyclopedia of personality and individual differences* (p. 55-60). John Wiley & Sons, Ltd. doi: 10.1002/9781119547174.ch186

Bahaei, S. S., & Gallina, B. (2021, September). A metamodel extension to capture post normal accidents in ar-equipped socio-technical systems. In *31st european safety and reliability conference.* doi: 10.3850/978-981-18-2016-8_417-cd

Ben Cheikh, A., Coulondre, S., Front, A., & Giraudin, J.-P. (2012). An engineering method for context-aware and reactive systems. In *2012 sixth international conference on research challenges in information science (rcis)* (p. 1-12). doi: 10.1109/RCIS.2012.6240433

Bocanegra, J., Pavlich-Mariscal, J., & Carrillo-Ramos, A. (2015). Dmlas: A domain-specific language for designing adaptive systems. In *2015 10th computing colombian conference (10ccc)* (p. 47-54). doi: 10.1109/ColumbianCC.2015.7333411

Brambilla, M., Cabot, J., & Wimmer, M. (2012). *Model-driven software engineering in practice* (Vol. 1). Morgan & Claypool Publishers. doi: 10.2200/S00441ED1V01Y201208SWE001

Brambilla, M., & Tziviskou, C. (2008). Modeling ontology-driven personalization of web contents. In *2008 eighth international conference on web engineering* (p. 247-260). doi: 10.1109/ICWE.2008.44

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571-583. (Software Performance) doi: 10.1016/j.jss.2006.07.009

Cabot, J. (2024). *The low-code handbook.* Jordi Cabot.

Conrardy, A. (2024). *User modeling in mde - slr data sheet and scripts.* doi: https://doi.org/10.5281/zenodo.14887422

DiMario, M., & Hodges, A. (2023). Systems engineering management in research and development valley of death. *INSIGHT*, *26*(3), 8-14. doi: https://doi.org/10.1002/inst.12451

Di Ruscio, D., Kolovos, D., Lara, J., Pierantonio, A., Tisi, M., & Wimmer, M. (2022, 01). Low-code development and model-driven engineering: Two sides of the same coin? *Software and Systems Modeling*, *21*. doi: 10.1007/s10270-021-00970-2

El Hog, C., Ben Djemaa, R., & Amous, I. (2012). Profile annotation for adaptable web service description. In *Proceedings of the 27th annual acm symposium on applied computing* (p. 1935–1940). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2245276.2232096

Elmagrouni, I., Lethrech, M., Kenzi, A., & Kriouile, A. (2016). Approach for building services-oriented systems adaptable. In *2016 5th international conference on multimedia computing and systems (icmcs)* (p. 183-188). doi: 10.1109/ICMCS.2016.7905656

Garrigós, I., Pardillo, J., Mazón, J.-N., Zubcoff, J., Trujillo, J., & Romero, R. (2012, October). A conceptual modeling personalization framework for olap. *J. Database Manage.*, *23*(4), 1–16. doi: 10.4018/jdm.2012100101

Gaspar, A., Gil, M., Panach, I., & Romero, V. (2024, 01). Towards a general user model to develop intelligent user interfaces. *Multimedia Tools and Applications*, *83*, 1-34. doi: 10.1007/s11042-024-18240-w

Grundy, J. C. (2021). Impact of end user human aspects on software engineering. In *Proceedings of the 16th international conference on evaluation of novel approaches to software engineering - enase* (p. 9-20). SciTePress. doi: 10.5220/0010531800090020

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., & von Wilamowitz-Moellendorff, M. (2005). Gumo: the general user model ontology. In *Proceedings of the 10th international conference on user modeling* (p. 428–432). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/11527886_58

Jaouadi, I., Ben Djemaa, R., & Ben-Abdallah, H. (2018, October). A model-driven development approach for context-aware systems. *Softw. Syst. Model.*, *17*(4), 1169–1195. doi: 10.1007/s10270-016-0550-0

Jaskolka, J., & Hamid, B. (2023). Towards the integration of human factors in collaborative decision making for secure architecture design. In *Proceedings of the 37th ieee/acm international conference on automated software engineering.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3551349.3561149

Jovanovic, M., Starcevic, D., & Jovanovic, Z. (2014). Bridging user context and design models to build adaptive user interfaces. In S. Sauer, C. Bogdan, P. Forbrig, R. Bernhaupt, & M. Winckler (Eds.), *Human-centered software engineering* (pp. 36–56). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kaklanis, N., Biswas, P., Mohamad, Y., Gonzalez, M. F., Peissner, M., Langdon, P., … Jung, C. (2016, March). Towards standardisation of user models for simulation and adaptation purposes. *Univers. Access Inf. Soc.*, *15*(1), 21–48. doi: 10.1007/s10209-014-0371-2

Karam, R., Fraternali, P., Bozzon, A., & Galli, L. (2012). Modeling end-users as contributors in human computation applications. In A. Abelló, L. Bellatreche, & B. Benatallah (Eds.), *Model and data engineering* (pp. 3–15). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kennedy, G., Shirvani, F., Scott, W., & Campbell, P. (2020). Managing organisational system evolution through model-based systems engineering approaches. In *2020 ieee international systems conference (syscon)* (p. 1-8). doi: 10.1109/SysCon47679.2020.9275920

Khalajzadeh, H., Grundy, J., & McIntosh, J. (2022). Vision: developing collaborative model-driven apps for personalised care plans. In *Proceedings of the 25th international conference on model driven engineering languages and systems: Companion proceedings* (p. 929–933). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3550356.3559100

Khider., H., Hammoudi., S., & Meziane., A. (2020). Business process model recommendation as a transformation process in mde: Conceptualization and first experiments. In *Proceedings of the 8th international conference on model-driven engineering and software development - modelsward* (p. 65-75). SciTePress. doi: 10.5220/0009155600650075

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Tech. Rep. No. EBSE 2007-001). Keele University and Durham University. (Joint Report)

Kohl, C., Mcintosh, E., Unger, S., Haddaway, N., Kecke, S., Schiemann, J., & Wilhelm, R. (2018, 02). Online tools supporting the conduct and reporting of systematic reviews and systematic maps: A case study on cadima and review of existing tools. *Environmental Evidence*, *7*. doi: 10.1186/s13750-018-0115-5

Liebel, G., Klünder, J., Hebig, R., Lazik, C., Nunes, I., Graßl, I., … Amaral, V. (2024, August). Human factors in model-driven engineering: future research goals and initiatives for MDE. *Software and Systems Modeling*, *23*(4), 801–819. doi: 10.1007/s10270-024-01188-8

Liu, S., Ma, C.-J., Chou, F.-Y., Cheng, M., Wang, C.-H., Tsai, C.-L., … Lu, T.-C. (2023). Applying a smartwatch to predict work-related fatigue for emergency healthcare professionals: Machine learning method. *Western Journal of Emergency Medicine: Integrating Emergency Care with Population Health*, *24*(4). doi: 10.5811/westjem.58139

Metin, H., & Bork, D. (2023). On developing and operating glsp-based web modeling tools: Lessons learned from biguml. In *2023 acm/ieee 26th international conference on model driven engineering languages and systems (models)* (pp. 129–139).

Michael, J., Bork, D., Wimmer, M., & Mayr, H. (2023, 10). Quo vadis modeling? *Software and Systems Modeling*, *23*, 1-22. doi: 10.1007/s10270-023-01128-y

Motti, V. G., & Vanderdonckt, J. (2013). A computational framework for context-aware adaptation of user interfaces. In *Ieee 7th international conference on research challenges in information science (rcis)* (p. 1-12). doi: 10.1109/RCIS.2013.6577709

Naveed, H., Arora, C., Khalajzadeh, H., Grundy, J., & Haggag, O. (2024). Model driven engineering for machine learning components: A systematic literature review. *Information and Software Technology*, *169*, 107423. doi: 10.1016/j.infsof.2024.107423

Nunes, I., Barbosa, S., Cowan, D., Miles, S., Luck, M., & Lucena, C. (2013, 01). Natural language-based representation of user preferences. *Interacting with Computers*, *27*, 133-158. doi: 10.1093/iwc/iwt060

Orellana, D. W., & Madni, A. M. (2014). Human system integration ontology: Enhancing model based systems engineering to evaluate human-system performance. *Procedia Computer Science*, *28*, 19-25. (2014 Conference on Systems Engineering Research) doi: 10.1016/j.procs.2014.03.003

Paige, R., & Cabot, J. (2024, 04). What makes a good modeling research contribution? *Software and Systems Modeling*, *23*, 1-5. doi: 10.1007/s10270-024-01177-x

Park, S., Nie, B. X., & Zhu, S.-C. (2018). Attribute and-or grammar for joint parsing of human pose, parts and attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(7), 1555-1569. doi: 10.1109/TPAMI.2017.2731842

Perrotin, P., Sadou, S., Hairion, D., & Beugnard, A. (2020). Detecting human vulnerably in socio-technical systems: a naval case study. In *Proceedings of the 23rd acm/ieee international conference on model driven engineering languages and systems: Companion proceedings.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3417990.3420045

Planas, E., Daniel, G., Brambilla, M., & Cabot, J. (2021, August). Towards a model-driven approach for multiexperience ai-based user interfaces. *Softw. Syst. Model.*, *20*(4), 997–1009. doi: 10.1007/s10270-021-00904-y

Plocher, T., Rau, P.-L. P., Choong, Y.-Y., & Guo, Z. (2021). Cross-cultural design. In *Handbook of human factors and ergonomics* (p. 252-279). John Wiley & Sons, Ltd. doi: 10.1002/9781119636113.ch10

Purificato, E., Boratto, L., & Luca, E. W. D. (2024). *User modeling and user profiling: A comprehensive survey.* doi: 10.48550/arXiv.2402.09660

Rich, E. (1979). User modeling via stereotypes. *Cognitive Science*, *3*(4), 329-354. doi: 10.1016/S0364-0213(79)80012-9

Robinson, L., Schulz, J., Blank, G., Ragnedda, M., Ono, H., Hogan, B., ... Khilnani, A. (2020, 07). Digital inequalities 2.0: Legacy inequalities in the information age. *First Monday*. doi: 10.5210/fm.v25i7.10842

Silva, S., & Belo, O. (2023). Domain and semantic modeling in the context of interactive systems development: User and device cases. In M. Papadaki, P. Rupino da Cunha, M. Themistocleous, & K. Christodoulou (Eds.), *Information systems* (pp. 497–514). Cham: Springer Nature Switzerland. doi: 10.1007/978-3-031-30694-5_36

Tapucu, D., Can, Ö., Bursa, O., & Ünalir, M. O. (2008). Metamodeling approach to preference management in the semantic web. In *Multidisciplinary workshop on advances in preference handling* (p. 116-123).

Tosi, M. D. L., Izquierdo, J. L. C., & Cabot, J. (2024). *A metascience study of the impact of low-code techniques in modeling publications.* doi: 10.48550/arXiv.2408.05975

Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R., & Conde, M. A. (2020). Representing data visualization goals and tasks through meta-modeling to tailor information dashboards. *Applied Sciences*, *10*(7). doi: 10.3390/app10072306

White, M., Clark, R. A. J., & Moore, J. D. (2010, 06). Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, *36*(2), 159-201. doi: 10.1162/coli.09-023-R1-08-002

Wischenbart, M., Mitsch, S., Kapsammer, E., Kusel, A., Pröll, B., Retschitzegger, W., ... Lechner, S. (2012). User profile integration made easy: model-driven extraction and transformation of social network schemas. In *Proceedings of the 21st international conference on world wide web* (p. 939–948). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2187980.2188227

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2601248.2601268

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., & Wessln, A. (2012). *Experimentation in software engineering*. Springer Publishing Company, Incorporated.

Yigitbas, E., Jovanovikj, I., Biermeier, K., Sauer, S., & Engels, G. (2020, September). Integrated model-driven development of self-adaptive user interfaces. *Softw. Syst. Model.*, *19*(5), 1057–1081. doi: 10.1007/s10270-020-00777-7

About the authors

Aaron Conrardy is a PhD student at the Luxembourg Institute for Science and Technology. His research focuses on leveraging software engineering techniques to democratize the development of adaptive user interfaces, seamlessly integrating AI technologies with everyday applications. You can contact the author at aaron.conrardy@list.lu or visit https://aaronconrardy.com.

Alfredo Capozucca received an MS degree in Computer Science from the National University of Rosario, Argentina (2003), and a PhD in Computer Science from the University of Luxembourg (2010). He is a full-time permanent research scientist in the Department of Computer Science at the Faculty of Science, Technology and Medicine at the University of Luxembourg. His current research interests lie at the intersection of modern driven engineering, formal methods, and sustainable computing. You can contact the author at alfredo.capozucca@uni.lu or visit https://acapozucca.github.io/.

Jordi Cabot is an FNR Pearl Chair and head of the Software Engineering RDI Unit at the ITIS department of the Luxembourg Institute of Science and Technology (LIST). Jordi is also an Affiliate Professor in CS at the University of Luxembourg. Previously, Jordi had been an ICREA Research Professor at Internet Interdisciplinary Institute, the Research center of the Open University of Catalonia (UOC) where Jordi led the SOM Research Lab. Jordi's research falls into the broad area of systems and software engineering, especially promoting the rigorous use of software models in all software tasks while keeping an eye on the most unpredictable element in any project, the people involved in it. Current research topics include pragmatic formal verification techniques, analysis of open source communities, open data exploitation and the role AI can play in software development (and vice versa). You can contact the author at jordi.cabot@list.lu or visit https://jordicabot.com.