arXiv:2505.24697v2 [cs.SE] 20 Jun 2026

This is a preprint.

The peer-reviewed version is published in the proceedings of the Engineering Interactive Computer Systems (EICS 2025) Workshop proceedings, as part of the third edition of the Engineering Interactive Systems Embedding AI Technologies workshop.

DOI: https://doi.org./10.1007/978-3-032-26051-2_7

TOWARDS A UNIFIED USER MODELING LANGUAGE FOR ENGINEERING HUMAN CENTERED AI SYSTEMS

Aaron Conrardy
Luxembourg Institute of Science and Technology
University of Luxembourg
Esch-sur-Alzette, Luxembourg
aaron.conrardy@list.lu

Alfredo Capozucca
University of Luxembourg
Esch-sur-Alzette, Luxembourg
alfredo.capozucca@uni.lu

Jordi Cabot
School of Coumputing and Information
Luxembourg Institute of Science and Technology
University of Luxembourg
Esch-sur-Alzette, Luxembourg
jordi.cabot@list.lu

ABSTRACT

In today's digital society, personalization has become a crucial aspect of software applications, significantly impacting user experience and engagement. A new wave of intelligent user interfaces, such as AI-based conversational agents, has the potential to enable such personalization beyond what other types of interfaces could offer in the past. Personalization requires the ability to specify a complete user profile, covering as many dimensions as possible, such as potential accessibility constraints, interaction preferences, and even hobbies. Yet, existing solutions for user modeling mostly focus on individual aspects at a very coarse level, severely limiting the potential adaptations for personalization. In this sense, this paper presents a unified user modeling language, aimed to combine previous approaches, both from the modeling community and other user-centric fields, in a single proposal. This language has been implemented on top of the open source BESSER low-code platform. Additionally, a proof of concept leveraging user profiles modeled with our language to automatically adapt a conversational agent has also been developed.

1 Introduction

Users are at the center of interactive applications [1]. Yet, each user is different and has different needs and expectations based on their specific profile (age, mood, personality, etc). Therefore, to provide an optimal user experience, applications need to adapt to each specific profile, providing a personalized interaction. Failing to do so could even lead to digital inequalities, as certain groups of users may not able to fully take advantage of an application due to their specific limitations (e.g. accessibility issues or language skills) [2].

Moreover, the growth of AI has increased the possibilities of user profiling and personalizing user interactions. Examples such as inferring the user's fatigue level based on smartwatch data using machine learning (ML) models [3] or predicting a user's behavior and attitudes based on an interview using Large Language Models (LLMs) [4] have now become possible. Furthermore, based on this profile data, AI also increases the degree of personalization options [5], such as automatically tuning text based on the language skills of the user [6] or performing user interface layout optimization based on user data [7].

To enable all these new opportunities, we need to be able to model such user profiles in a structured and machine-readable manner, which is known as user modeling [8]. Nevertheless, there is not yet a general user modeling language [9, 10] that could be used as the *de facto* standard for the model-driven engineering (MDE) community and beyond.

Conversely, there are myriad of partial and overlapping solutions that result in incomplete profiles and interoperability issues.

This paper aims to provide such user modeling language by collecting and unifying previous proposals from the modeling community, complemented with user profiling perspectives from other fields (sociology, psychology, etc.). Additionally, to illustrate the benefits of our language for personalization purposes, we have implemented a proof of concept that automatically adapts the behavior of an LLM agent based on a given user profile.

The rest of the paper is structured as follows: Section 2 describes the state of the art on user modeling languages and current drawbacks. Section 3 presents the metamodel of the proposed unified user modeling language and discusses a possible concrete syntax. The implemented proof of concept is described in Section 4 and the corresponding tool support in Section 5. Finally, we present future work in Section 6 and conclude the paper in Section 7.

2 State of the art

We have conducted a systematic literature review (SLR) on user modeling in MDE [10]. The SLR comprised 30 papers that proposed at least some kind of abstract formalization of a user modeling language (such as a metamodel or grammar) and investigated its integration into engineering pipelines. In what follows, we briefly discuss the most relevant results.

As stated before, there is no unified user modeling language. Instead, most works rather attempted to create a new proposal from scratch, ignoring existing results, and thus, leading to a fragmented community that proposes overlapping user dimensions while, at the same time, missing others already proposed in previous works. Additionally, there is a tendency to favor more static user aspects, while neglecting more complex dynamic ones (like mood or fatigue) and/or present the dimensions at a very high level without giving enough details on the possible values that could be set for each dimension. A possible explanation for these limitations is that, in most of such works, modeling the user is a secondary concern rather than a first-class citizen, missing out on potential personalization opportunities.

Out of the reviewed papers, there were two exceptions that had a goal similar to ours. Kaklanis et al. [11] aimed to provide a standard of user models for the purpose of user simulation in virtual environments. They explicitly mention that their focus was on an interoperable user model to describe able-bodied individuals and people with various kinds of disabilities. Thus, the scope is rather narrow compared with ours. Similarly, Gaspar et al. [12] created a unified user model, this time specifically for dimensions that directly affect the user interface to be presented to the user. Again, our target language aims to be more generic and enable all types of user adaptations. Even more importantly, the dimensions defined by Gaspar et al. are not detailed and stay at a very generic level which hinders the reusability of this approach.

3 A unified user modeling language

To overcome the fragmented nature of user models and provide a more complete, while at the same time, fine-grained, representation of users, we developed a new language that integrates insights from the SLR and user dimensions from other fields. This section presents the abstract and concrete syntax of the unified user modeling language.

3.1 Abstract syntax: The unified user metamodel

We begin by defining the abstract syntax of the user modeling language, formalized as a metamodel and represented, as usual, using the UML class diagram notation. As starting point for the metamodel, we took the results from the conducted SLR [10] as it provided a list of user dimensions and a classification for these. We have then extended such list of dimensions, e.g. by proposing the additional category "Culture", as it is regarded as another important source of influence of the user [13, 1], but was missing from the reviewed literature. In some cases, dimensions are shared by multiple categories (e.g. age fits in both "Personal Information" and "Accessibility"), yet we present them as part of the most fitting one for convenience.

Figure 1 depicts a high-level view of the metamodel, showcasing the main categories that characterize a user. Note that all categories (and the dimensions within) are optional as users may decide not to disclose that information or, in automatic profiling methods, we may not know them from the beginning.

Furthermore, as many dimensions include a range value, we introduce a new datatype with the same name to represent integer numbers in the range of 0 to 100. In most cases, this datatype is used when a dimension represents a score, such as the level of happiness a user is currently experiencing. Some dimensions instead compare two opposing concepts. In those cases, the Range is used as a scale following the naming scheme "X_To_Y" to compare X to Y, such as "bad_to_good" (Figure 7) that represents whether a user is leaning towards a bad (value leaning towards 0) or good

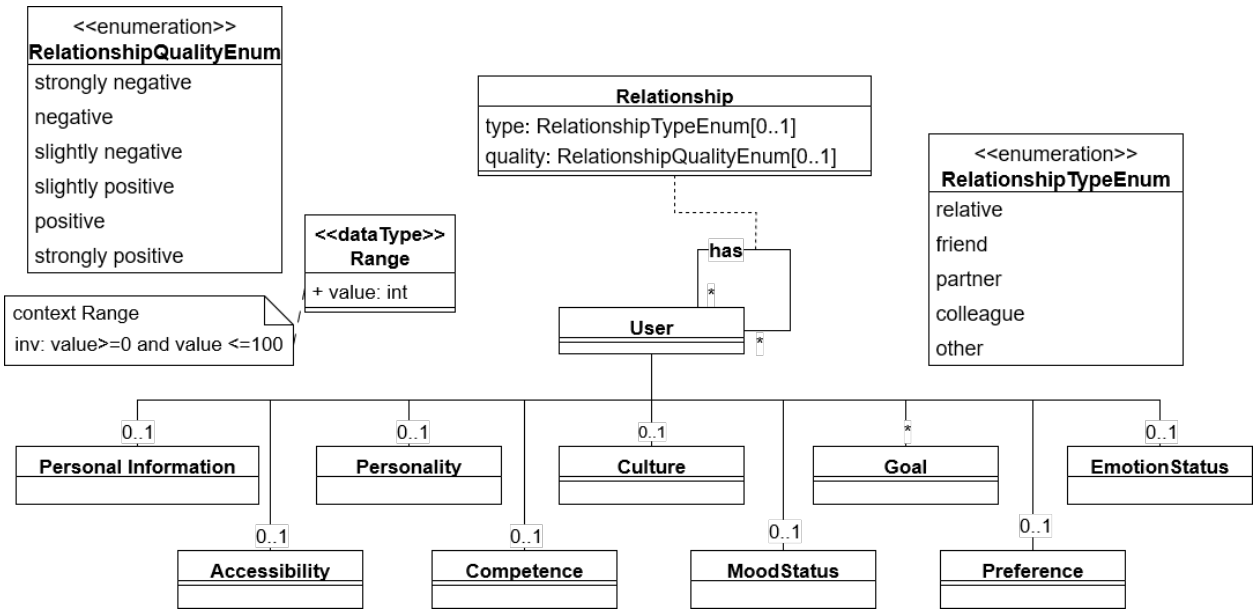


Figure 1: High-level view of unified user metamodel

(value leaning towards 100) mood. Range generally offers flexibility, as depending on the requirements, one could map a predefined selection of choices to the values of Range (e.g. a binary selection of "No/Yes" could be mapped to the values 0 and 100 of Range), simulating an enumeration without altering the metamodel.

In the following, we will go over the main concepts of the metamodel, including all categories and briefly discuss their content. Due to size limitations, parts of the different metamodel are presented condensed. A complete version is available online.[1] The online version contains the complete metamodel, detailed information on the user dimensions and a mapping of reference to user dimension.

3.1.1 Relationship

Information on user relationships with other users is popular in research works that deal with user models for social network platforms (e.g. [14]). However, most models only provide information on the existence of a relationship but not the type of relationship or any other relevant data. We propose the possibility to also capture the type of relationships (relative, partner, colleague, etc.) and the quality of the relationship (Relationship class in Figure 1). This information is valuable in understanding the nuance of a relationship, as not every relationship is of positive nature and it may affect processes such as content recommendation.

3.1.2 Personal information

Personal information (Figure 2) describes typical information a user would enter on a profile page when registering for a new social application (e.g. a social network account). That includes information such as the full name, age, address or gender of the user.

Regarding "Interest" and "Hobby", while both deal with engaging in topics via an activity, "Hobby" tackles a more recurrent and dedicated engagement. Therefore, we model "Hobby" as a stronger version of "Interest" where we indicate the number of weekly hours the user devotes to the hobby. The "Topic" class represents a non-exhaustive list of topics, derived from an ontology provided in [15].

3.1.3 Competence

Competence (Figure 3) describes a user's proficiency on a number of aspects such as education, language or a various number of skills and expertises. Regarding languages, we propose the usage of the Common European Framework of Reference for Languages[2] (CEFR) as a standard description of a user's language skills. Regarding "Skill" and

[1] `https://github.com/BESSER-PEARL/User-Modeling-Language`

[2] https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions

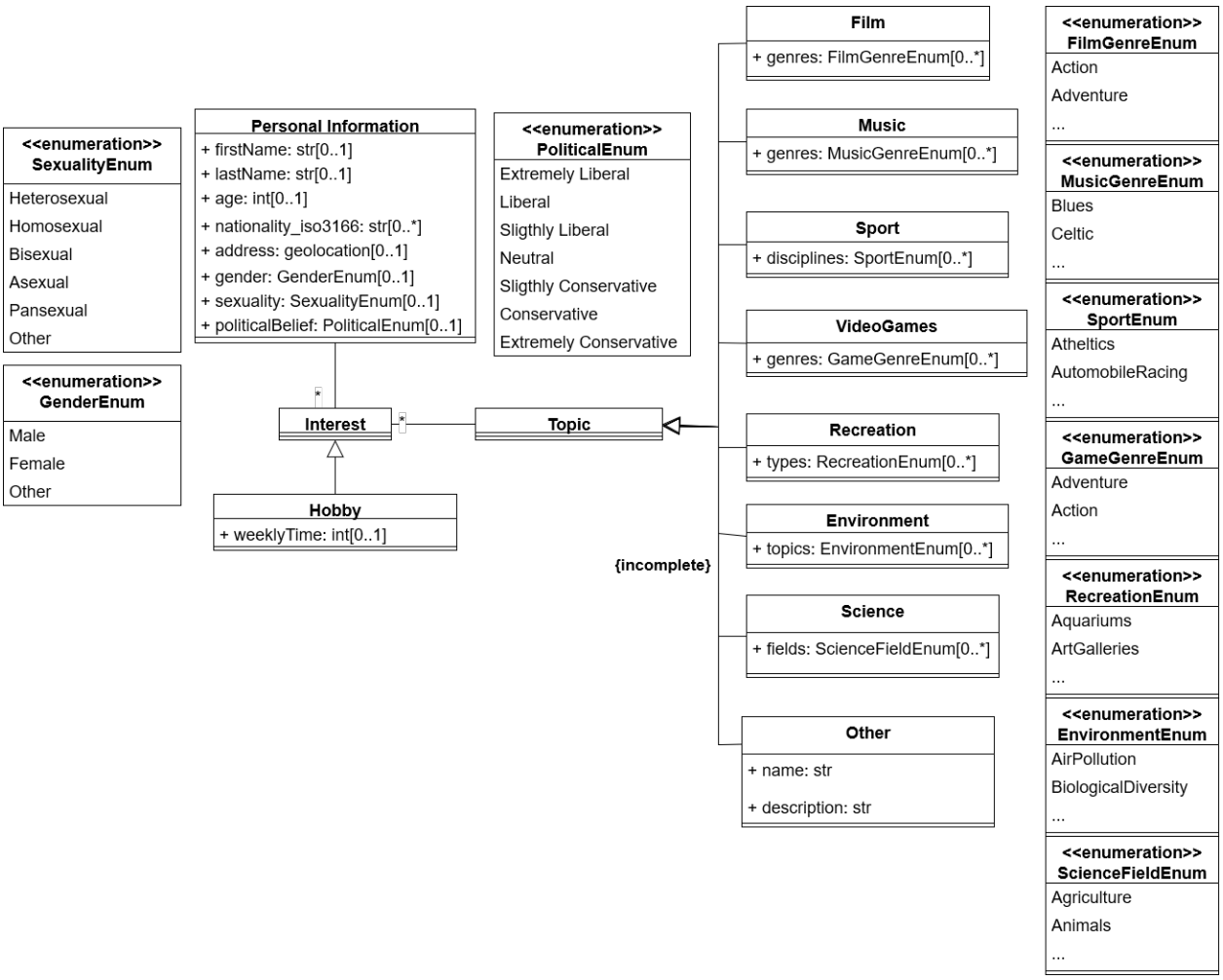


Figure 2: Personal information

"Knowledge", the attribute "score" is used to describe the user's ability to perform a skill or level of knowledge respectively. For the latter, "Knowledge" relates to the topics contained in the previously presented "Topic" class.

3.1.4 Accessibility

Accessibility (Figure 4) refers to the accessibility needs of the user. The "Disability" class describes a concrete disability a user might have, whereas the other classes ("Mobility", "Speech", "Hearing", etc.) describe the state of the different senses, parts of the body and body functions. Of course, Disability and the other classes affect each other, as a completely deaf person will have the lowest hearing values. While most of the classes are inspired by Kaklanis et al. [11], their metamodel did not contain any attribute types. Therefore, we completed the metamodel by providing attribute types based on the used values from the results of common measurement methods for a bodily function. An example would be the usage of decibels to measure a person's hearing during an audiogram [16], making the choice of integers fitting.

3.1.5 Personality

Personality (Figure 5) tackles the user's personality traits, that is, the stable internal characteristics of the users. Numerous personality models were developed by psychologists to explain the nature of humans and their differences. For our purposes, we opted to cover personality traits from the most popular models (as proposed in [15]), such as the Big Five personality traits [17]. These personality traits, represented by the "Trait" class, describe people's pattern of thoughts, feelings and behaviors [18], while the "Characteristic" class describes observable and concrete feature of a person implied by a trait. Both classes have a score to denote the extent to which a user fulfills a trait/characteristic.

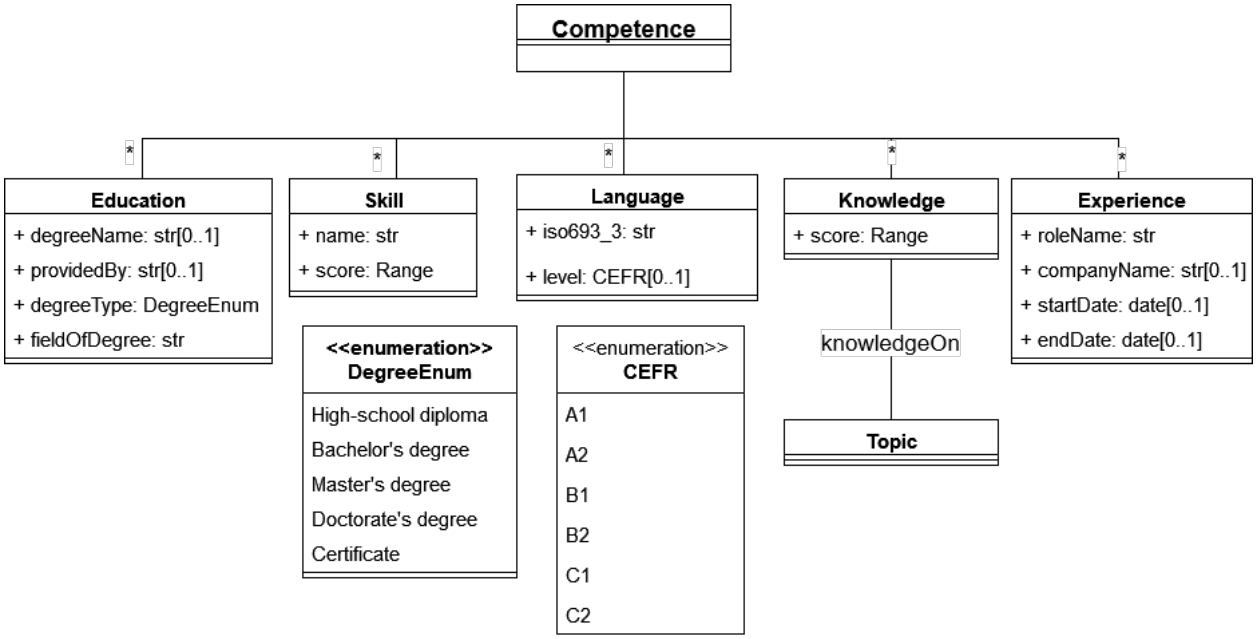


Figure 3: Competence

Additional dimensions affected by, or linked to personality were included, such as Motivation and Attitude. The former is based on a unified motivational model proposed by Forbes [19] that explains the different motivational constructs a person might lean towards to. The latter tackles a person's opinion given a topic, be it of a negative or positive nature. Bias in that sense is a stronger and more negative version of Attitude, often leading to an unfair treatment of a target.

3.1.6 Preference

Preferences (Figure 6) are used to represent a user's personal preferences, that is, a preferred option in regards to a set of choices. In the context of a software application, the choices are narrowed to preferences of the content itself (such as preferring one topic over another or the preferred language), design preferences (such as a preferred color scheme) and the preferred interaction modality (such as audio or textual interaction).

3.1.7 Culture

In general, culture (Figure 7) is linked to the social behavior, institutions and norms found in human societies and is often attributed to a specific region/location. To model the culture of a user, we followed Hofstede's cultural dimensions [20]. These consist of dimension pairs that are opposed to each other (e.g. Collectivism and Individualism), which is why we used our Range datatype. Additionally, we added Religion as a dimension, as it is often linked to the culture of regions/locations and influences expectations during user interaction with an application [21].

3.1.8 Goals

Goals (Figure 7) focus on the objectives of a user, both while interacting with an application and outside. Optionally, a goal can have a deadline. Essentially, goals provide a purpose behind the user's actions and can be both static or dynamic depending on the circumstances (a goal could appear based on an interaction that just took place).

3.1.9 Emotions and Moods

Similarly to personality, there are various models in psychology that attempt to quantify emotions and moods (Figure 7). Regarding the concrete difference between emotions and moods, emotions tend to be associated with a specific object or event and do not last long, while moods tend to last longer and have a less direct cause [22]. For emotions, we decided to showcase the list of emotions as presented in GUMO [23]. For moods, we opted for the three pairs of opposing moods as presented in [24].

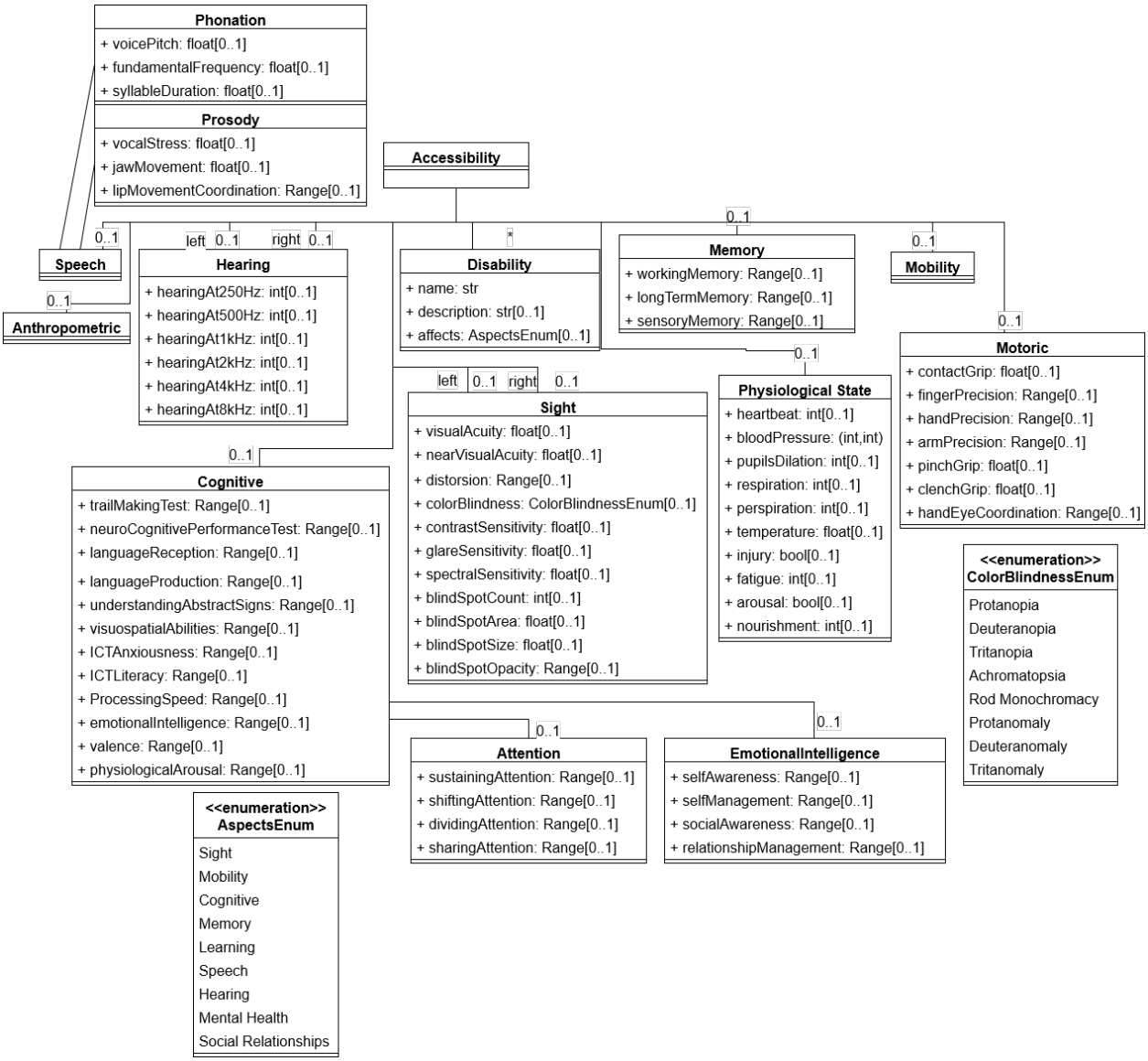


Figure 4: Accessibility

3.2 Concrete syntax

Based on the presented abstract syntax, we derived a textual concrete syntax to facilitate the creation of user models. In particular, we have chosen a JSON-based notation due to its popularity and its human- and machine-readable nature, which facilitates the use of user models in AI-driven personalization scenarios. Indeed, LLMs were already shown to be able to understand [25] as well as produce JSON [26]. Beyond LLMs, many tools work with JSON documents which opens the door to numerous applications exploiting user models.

JSON additionally allows for the definition of a JSON schema[3] that enables data consistency and validity checks of given JSON data based on predefined rules. In our case, the JSON Schema defines the concepts in the metamodel, and user models are represented as JSON objects that conform to this schema. Figure 8a shows an excerpt of the metamodel in JSON schema format whereas Figure 8b showcases the use of the concrete syntax to write user profile data that can be validated using the defined schema.

[3]https://json-schema.org/

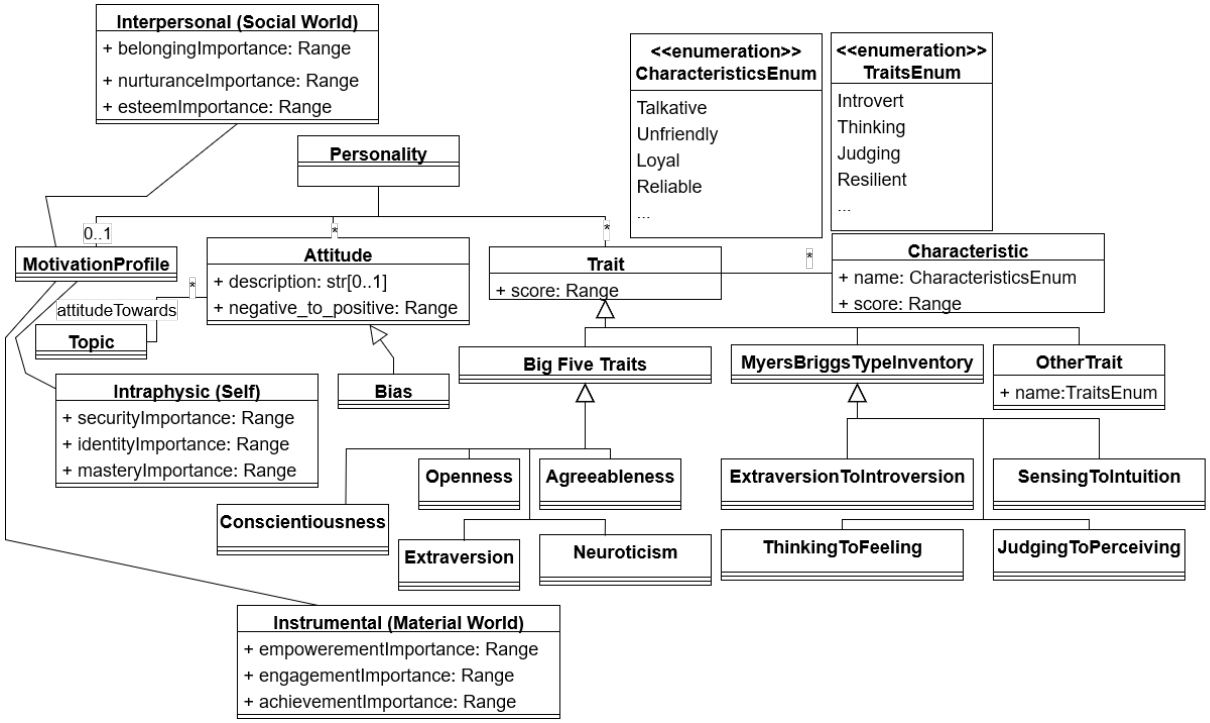


Figure 5: Personality

In Section 5, we demonstrate how JSON objects representing the user model can be automatically generated from a form-based web application, further simplifying their creation.

4 Proof of concept: Automatic conversational agent personalization

To showcase the usefulness of the user modeling language, we have implemented a personalization pipeline that, given an input user model, automatically personalizes the responses of a conversational agent to increase the user experience and engagement.

Our pipeline supports two approaches to personalize conversational agents (depicted in Figure 9), depending on the capabilities and access we have to the target LLM powering the conversational agent:

- Direct personalization of LLM-agent: The information of the user model is directly embedded into the LLM via fitting context prompts. As a result, the conversation responses generated by the LLM are inherently personalized.
- Indirect personalization of the conversational agent: When we have an non-personalized message, whether it comes from the LLM without user context or is pre-defined by a developer, we post-process it. We do this by sending the original message and the user model to a secondary LLM, which produces a personalized version.

In the following, we describe the first approach. The process of the second approach is similar, except that it involves a separate request to an LLM with a slightly different prompt. We propose this second approach for two reasons: (1) it allows using one LLM optimized for text generation and another optimized for personalization, and (2) it supports a closed conversational agent with pre-defined conversation paths, preventing the user from engaging in open-ended dialogue.

Initially, a web application provides an interface for end-users to interact with a non-personalized LLM, with the option to upload a user model for personalization. Once a user model that follows the JSON schema as described in Section 3.2 is uploaded, the following prompt context is automatically added to the conversation request "You adapt your responses based on a given user profile, such as their native language, interests, and other provided attributes. Your goal is to enhance the user's experience by tailoring your responses to their profile. This is the user's profile {JSON file containing User Model}". As mentioned before, we take advantage of LLM's capabilities to understand JSON and therefore omit the transformation of the user model to a natural language form.

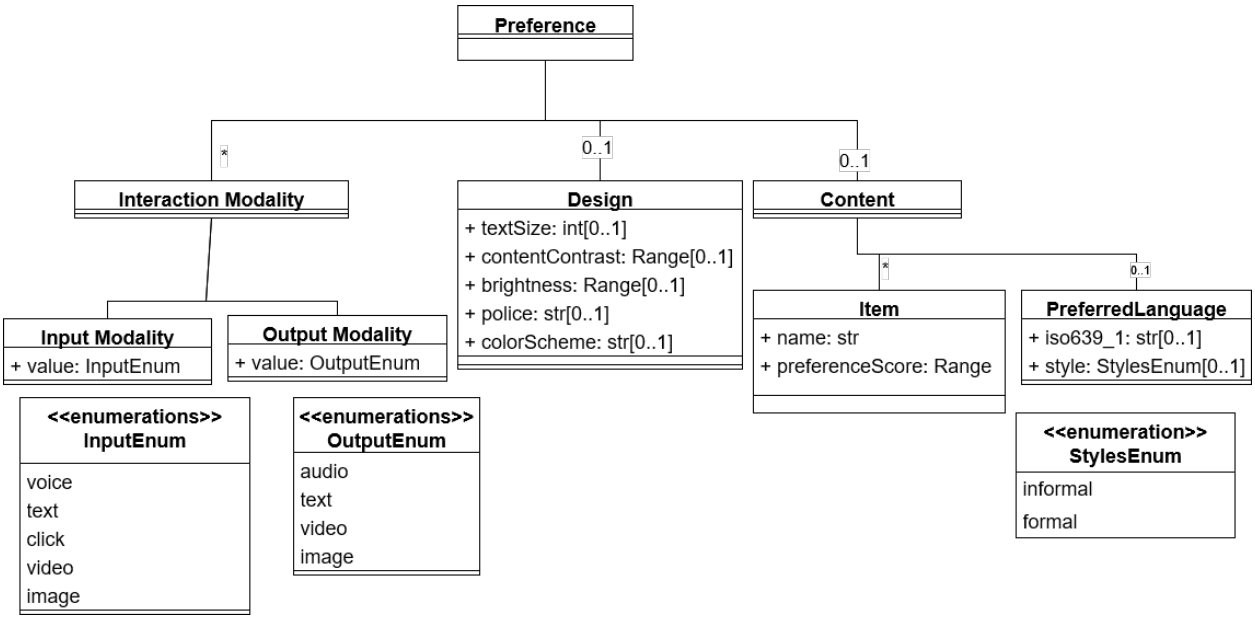


Figure 6: Preference

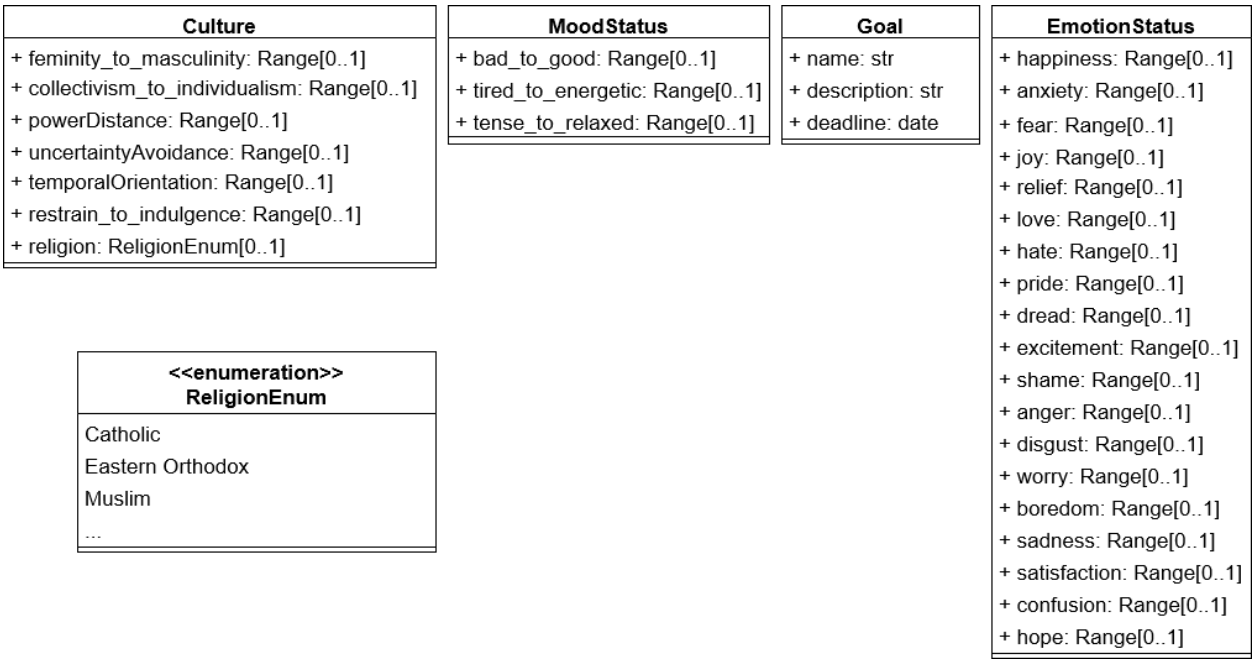


Figure 7: Culture, Mood, Goal and Emotion

Once this sub-prompt is ready, it is automatically embedded into an LLM's context information. In our case, we used OpenAI's[4] GPT-4o-mini model due to its current reputation as one of the most performant LLM [27]. Using OpenAI's API[5], one can set context information at the system level, leading to an improved upholding of defined context rules.

Finally, the web application updates its layout to indicate that the user is now interacting with a personalized LLM.

As an example, Figure 10 shows the initial web application and an example interaction with GPT-4o-mini, whereas Figures 11a and 11b contain a comparison of the same question being asked with different user models stored in its

[4]https://openai.com/
[5]https://platform.openai.com/docs/overview



(a) JSON schema



(b) Instantiation following concrete syntax

Figure 8: Example use of JSON syntax following the JSON schema

context. Specifically, we prompted the LLM to generate exercise recommendations for muscle gain, constrained to three sentences, reflecting the use case of an AI coach providing beginner-friendly gym advice while considering the user's profile. We notice an adapted response based on the user's age or disability, showcasing the LLM's capabilities to adapt its responses based on the given combination of prompt and user model. We argue that in this simple example, the personalized messages improve the user experience for the given profiles. Naturally, a user study would be necessary to confirm this. Such a study could consist of showing the default and personalized responses and letting participants evaluate their preferred answer.

5 Tool support

The user modeling language and the proof of concept were implemented using existing and extended components from the open-source BESSER[6] low-code platform [28] and are both available online[7].

[6]https://github.com/BESSER-PEARL/BESSER

[7]`https://github.com/BESSER-PEARL/User-Modeling-Language`

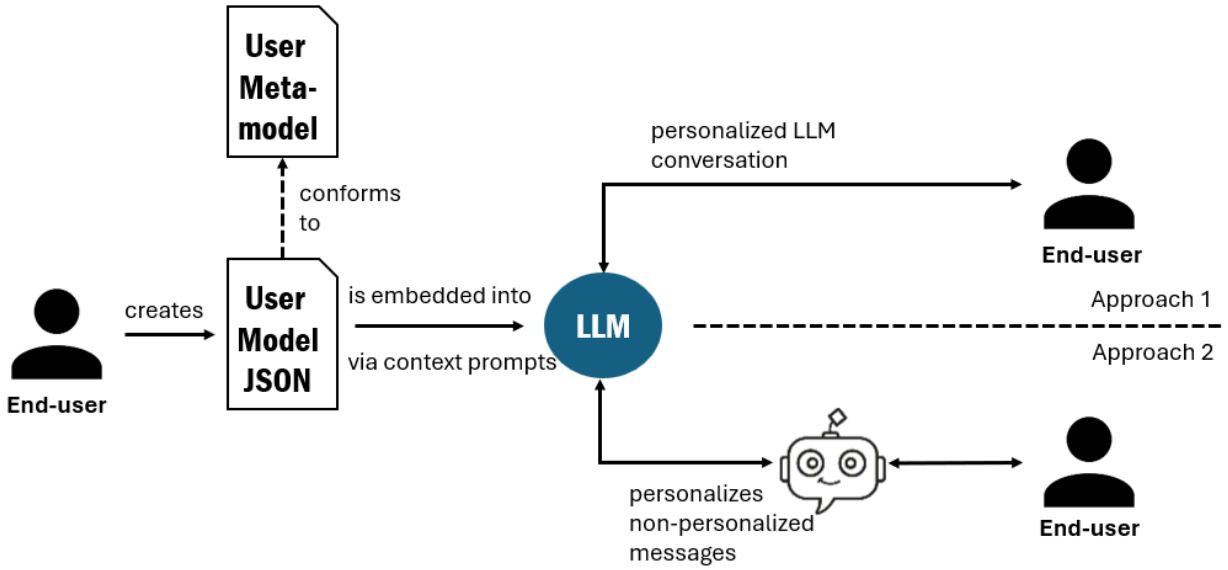


Figure 9: Implemented approaches to personalize conversational agent

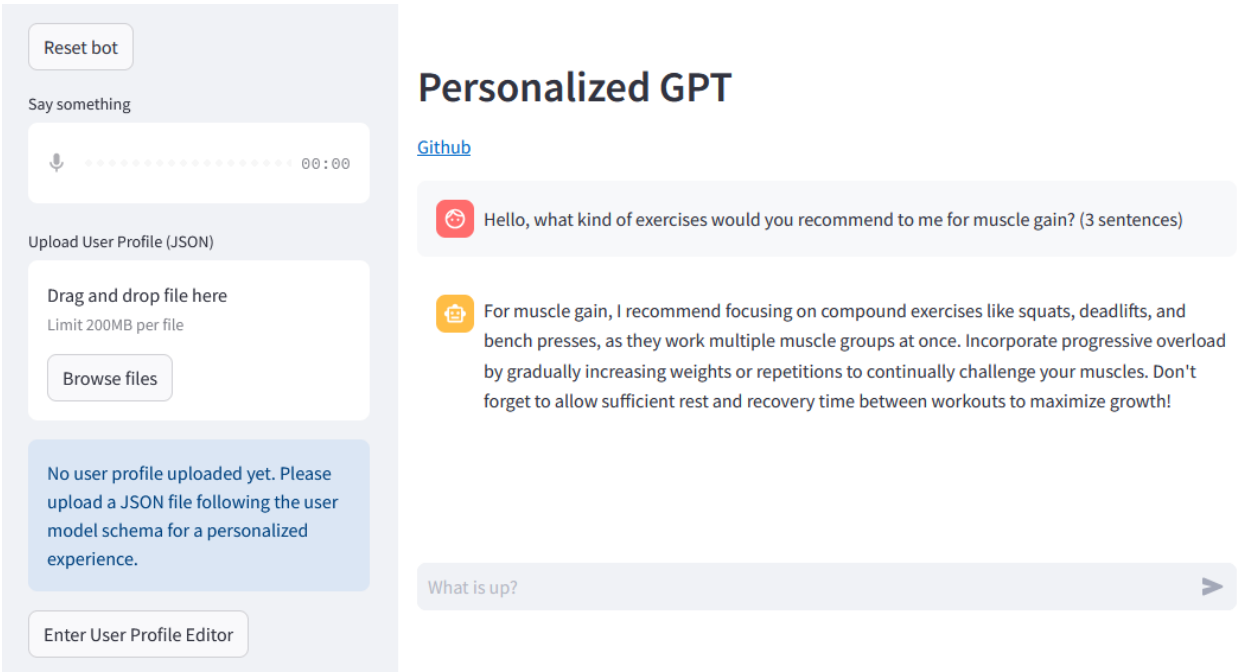
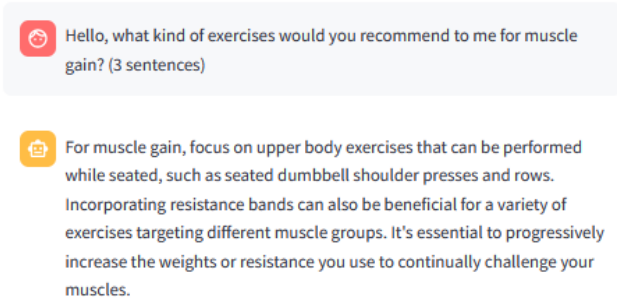
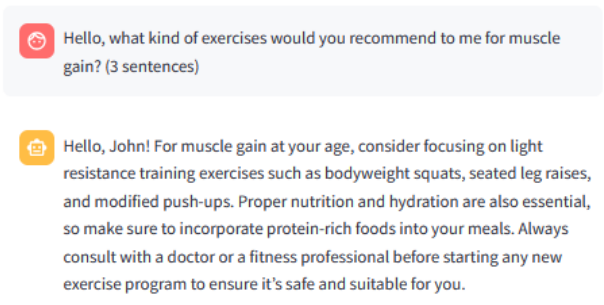


Figure 10: Overview of chat interface and example conversation with LLM (GPT-4o-mini) without personalization



(a) Profile of paraplegic male aged 30: We notice the omission of exercises involving the lower body and an increased focus on upper body exercises



(b) Profile of male aged 80: We notice a focus on light exercises, avoiding weighted exercises and recommendations to visit a doctor to be safe due to the age

Figure 11: Interaction with personalized LLM (GPT-4o-mini) using different user models

For the user modeling language, we first used BESSER's existing modeling language called B-UML to model the abstract syntax as a structural model following its textual syntax. We developed a new generator on top of the BESSER generators that performs a model to text transformation, transforming an input structural model into the equivalent JSON schema as defined in Section 3.2. The advantage of this generator is the possibility to easily update the abstract syntax and immediately updating the JSON schema. This property enables future contributors to easily extend the user modeling language.

To facilitate the usage of the concrete syntax, we developed a Streamlit[8] web-based application that lets end-users create a user model by simply filling a form, also available on the repository of the user modeling language. Following the defined cardinalities, a user has the option to only fill out a part of the form and can leave some information out. Figure 12 shows an excerpt of the personal information forms. Once filled out, the information is mapped to the concrete syntax and stored as a JSON file.

The personalizable LLM was developed using BESSER's agentic framework[9]. The framework contains wrappers for different LLMs and allows for the definition of context information that will be provided to the LLMs as high-priority context information. Additionally, BESSER proposes the previously shown frontend (Figure 10) to interact with the created personalized LLM-agents. Both the forms and conversational agents frontend are combined, such that one can enter the forms page via the conversational agent frontend ("User Profile Editor" button in Figure 10), enter the profile information and start the conversation with the personalized agent, removing the need for the end-user to manually import the user model.

6 Future work

6.0.1 Interleaving with existing application modeling languages to engineer AI systems

The proposed proof of concept does not allow to define which kind of adaptation takes place, but rather, acts non-deterministically and performs automatic adaptation outside of our control. The defined user modeling language should be combined with application modeling languages (e.g. IFML[10]), specifying different kinds of behaviors based on specific values. The resulting combined language should allow for specifying both automatic and pre-defined adaptations, taking the best of both worlds. Existing attempts at combining user profile information and application behavior, such as [29], could act as an inspiration and be extended, avoiding re-inventing the wheel. Correspondingly, generators shall be implemented that ingest the models and produce the adaptive application, enabling a pipeline that allows developers to both design and automatically produce (parts of) a user centered AI system.

Additionally, each language should be equipped with a graphical concrete syntax, increasing the user experience for the developers, providing straightforward drag and drop components leading to an easy and fast development.

[8] https://streamlit.io/
[9] https://github.com/BESSER-PEARL/BESSER-Agentic-Framework
[10] https://www.ifml.org/

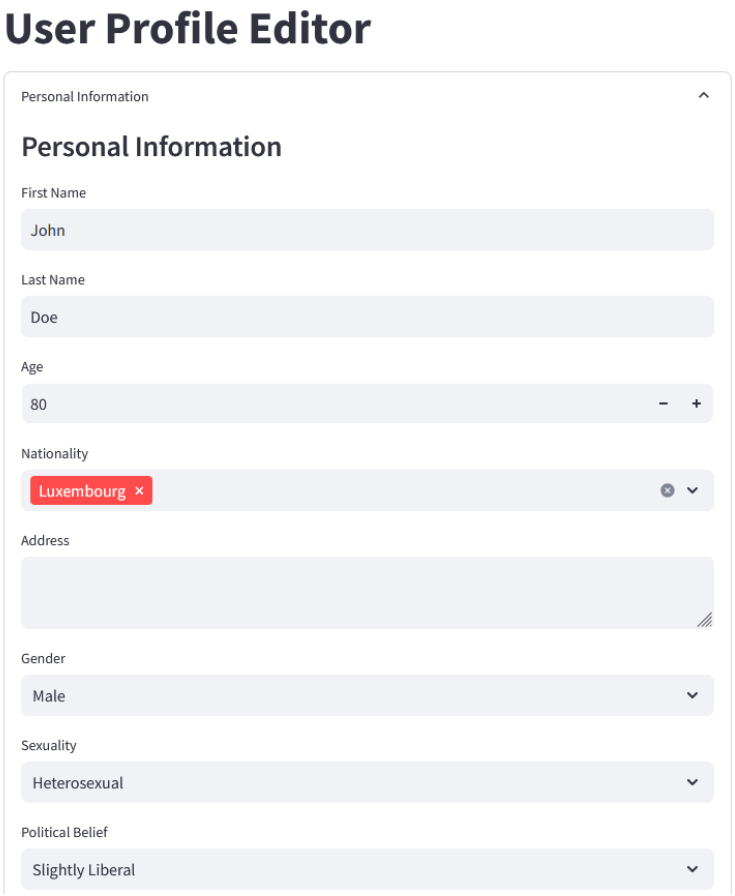


Figure 12: Forms frontend

6.0.2 Quality evaluation of user models

Performing quality evaluations of created user models using model-based testing, verification and validation approaches is mostly unexplored [10]. Existing approaches used for application models could be adapted to be applied for user models. Techniques to check the user model's consistency (e.g. a deaf user should not declare audio as a preferred modality) or verifying that pre-defined rules are evaluated to true (e.g. the age of a user cannot be over 200) should be included in the user modeling language.

6.0.3 Automatic profiling of user

Beyond static information to be provided by the end-user, we will explore how to support dynamic scenarios where applications need to adapt to users for which there is no profile information available. For such users the idea would be to discover a basic profile through the interactions with the systems to, slowly but steadily, adapt the interface to their profile. Existing methods (e.g. emotion recognition using [29] computer vision) and emerging ones (e.g. inferring the user's behaviors and attitudes based on a conversation with an LLM-agent [4]) need to be explored to allow the application to create an accurate model of the user via explicit and implicit methods.

6.0.4 Ethical and Privacy Considerations

The processing of personal data is governed by legal and ethical requirements, most notably the General Data Protection Regulation (GDPR). Consequently, any personalization framework must be designed to ensure compliance with these requirements. Key aspects include obtaining and managing user consent, minimizing the collection of unnecessary data, limiting data retention to what is strictly required for processing, and pseudonymizing data when delegating processing to third parties.

7 Conclusion

In this paper, we have proposed a unified user modeling language that is the sum and adaptation of fragmented solutions from the MDE community and beyond. First, the abstract syntax was formalized as a metamodel that covers a wide range of clearly defined user dimensions, allowing for a complete representation of any user and increasing the personalization potential. Secondly, the concrete syntax is based on a JSON schema that was derived from the metamodel and enables the creation of user models that capture the users' information. Compared with the reviewed works [10], our language is much more complete, as the maximum number of categories (respectively user dimensions) covered by a solution in previous works was 5 (respectively 14), while our metamodel covers 10 categories and provides more granular details on the proposed dimensions, clearly providing a more exhaustive palette to represent users.

Moreover, the proof of concept showcases the usefulness of the user modeling language by using the produced user models to automatically personalize a conversational agent. We argue that the provided adaptation improves the quality of the response for the given profile and, consequently, the overall user experience.

Implementing the points defined in future work will establish a pipeline to model, test and generate human centered AI systems, speeding up their development while ensuring compliance with existing regulations.

References

[1] John C. Grundy. 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*, pages 9–20. INSTICC, SciTePress, 2021.

[2] Laura Robinson, Jeremy Schulz, Hopeton Dunn, Antonio Casilli, Paola Tubaro, Rod Carveth, Wenhong Chen, Julie Wiest, Mati Dodel, Michael Stern, Christopher Ball, Kuo-Ting Huang, Grant Blank, Massimo Ragnedda, Hiroshi Ono, Bernie Hogan, Gustavo Mesch, Susan Kretchmer, and Aneka Khilnani. Digital inequalities 3.0: Emergent inequalities in the information age. *First Monday*, 07 2020.

[3] S. Liu, C. Ma, F. Chou, M. Cheng, C. Wang, C. Tsai, et al. 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), 2023.

[4] Joon Sung Park, Carolyn Q. Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S. Bernstein. Generative agent simulations of 1,000 people, 2024.

[5] Elena Planas, Gwendal Daniel, Marco Brambilla, and Jordi Cabot. Towards a model-driven approach for multiexperience ai-based user interfaces. *Softw. Syst. Model.*, 20(4):997–1009, August 2021.

[6] Emiliana Murgia, Maria Soledad Pera, Monica Landoni, and Theo Huibers. Children on chatgpt readability in an educational context: Myth or opportunity? In *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '23 Adjunct, page 311–316, New York, NY, USA, 2023. Association for Computing Machinery.

[7] Yue Jiang, Zixin Guo, Hamed Rezazadegan Tavakoli, Luis A. Leiva, and Antti Oulasvirta. Eyeformer: Predicting personalized scanpaths with transformer-guided reinforcement learning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA, 2024. Association for Computing Machinery.

[8] Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. User modeling and user profiling: A comprehensive survey, 2024.

[9] Grischa Liebel, Jil Klünder, Regina Hebig, Christopher Lazik, Inês Nunes, Isabella Graßl, Jan-Philipp Steghöfer, Joeri Exelmans, Julian Oertel, Kai Marquardt, Katharina Juhnke, Kurt Schneider, Lucas Gren, Lucia Happe, Marc Herrmann, Marvin Wyrich, Matthias Tichy, Miguel Goulão, Rebekka Wohlrab, Reyhaneh Kalantari, Robert Heinrich, Sandra Greiner, Satrio Adi Rukmono, Shalini Chakraborty, Silvia Abrahão, and Vasco Amaral. Human factors in model-driven engineering: future research goals and initiatives for MDE. *Software and Systems Modeling*, 23(4):801–819, August 2024.

[10] Aaron Conrardy, Alfredo Capozucca, and Jordi Cabot. User modeling in model-driven engineering: A systematic literature review. *Journal of Object Technology*, 24(2):2:1–14, May 2025. The 21st European Conference on Modelling Foundations and Applications (ECMFA 2025).

[11] N. Kaklanis, P. Biswas, Y. Mohamad, M. F. Gonzalez, M. Peissner, P. Langdon, D. Tzovaras, and C. Jung. Towards standardisation of user models for simulation and adaptation purposes. *Univers. Access Inf. Soc.*, 15(1):21–48, March 2016.

[12] Alberto Gaspar, Miriam Gil, Ignacio Panach, and Verónica Romero. Towards a general user model to develop intelligent user interfaces. *Multimedia Tools and Applications*, 83:1–34, 01 2024.

[13] Tom Plocher, Pei-Luen Patrick Rau, Yee-Yin Choong, and Zhi Guo. *Cross-Cultural Design*, chapter 10, pages 252–279. John Wiley & Sons, Ltd, 2021.

[14] Roula Karam, Piero Fraternali, Alessandro Bozzon, and Luca Galli. Modeling end-users as contributors in human computation applications. In Alberto Abelló, Ladjel Bellatreche, and Boualem Benatallah, editors, *Model and Data Engineering*, pages 3–15, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[15] Dominik Heckmann. *Ubiquitous User Modeling*. 01 2006.

[16] Roy D. Patterson, Ian Nimmo-Smith, Daniel L. Weber, and Robert Milroy. The deterioration of hearing with age: Frequency selectivity, the critical ratio, the audiogram, and speech threshold. *The Journal of the Acoustical Society of America*, 72(6):1788–1803, 12 1982.

[17] Anita Feher and Philip A. Vernon. Looking beyond the big five: A selective review of alternatives to the big five model of personality. *Personality and Individual Differences*, 169:110002, 2021. Celebrating 40th anniversary of the journal in 2020.

[18] Jorden A. Cummings and Lee Sanders. *Introduction to Psychology*. University of Saskatchewan Open Press, Saskatoon, SK, 2019.

[19] David L. Forbes. Toward a unified model of human motivation. *Review of General Psychology*, 15(2):85–98, 2011.

[20] Geert Hofstede. Dimensionalizing cultures: The hofstede model in context. *International Journal of Behavioral Medicine - INT J BEHAVIORAL MEDICINE*, 2, 01 2007.

[21] Aaron Marcus. User interface design and culture. *Usability and internationalization of information technology*, 3:51–78, 2005.

[22] Andrew Lane, Christopher Beedie, and Peter Terry. Distinctions between emotion and mood. *Cognition and Emotion*, 19, 09 2005.

[23] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta Wilamowitz-Moellendorff. Gumo – the general user model ontology. pages 428–432, 07 2005.

[24] Katharina Lochner and M. Eid. *Successful emotions: How emotions drive cognitive performance*. 01 2016.

[25] Kaisa Svennberg and Jan Ekman. Structuring semi-structured data from building inspection reports using a large language model. In *Multiphysics and Multiscale Building Physics*, pages 508–513, Singapore, 2025. Springer Nature Singapore.

[26] Michael J. Mior. Large language models for json schema discovery, 2024.

[27] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-free llm benchmark, 2024.

[28] Iván Alfonso, Aaron Conrardy, Armen Sulejmani, Atefeh Nirumand, Fitash Ul Haq, Marcos Gomez-Vazquez, Jean-Sébastien Sottet, and Jordi Cabot. Building besser: An open-source low-code platform. In *Enterprise, Business-Process and Information Systems Modeling*, pages 203–212, Cham, 2024. Springer Nature Switzerland.

[29] Enes Yigitbas, Ivan Jovanovikj, Kai Biermeier, Stefan Sauer, and Gregor Engels. Integrated model-driven development of self-adaptive user interfaces. *Softw. Syst. Model.*, 19(5):1057–1081, September 2020.