User Metrics for Adaptive Interface Design

Alberto Gaspar[1*], José Ignacio Panach[1], Miriam Gil[1], Jean Vanderdock[2]

[1*]Universitat de València, Av. de la universitat s/n, Burjassot, 46100, Valencia, Spain.
[2]Université Catholique de Louvain, Louvain School of Management, Pl. de l'Université 1, Louvain-la-Neuve, 1348, Walloon, Belgium.

*Corresponding author(s). E-mail(s): algasvi@alumni.uv.es;
Contributing authors: joigpana@uv.es; miriam.gil@uv.es;
jean.vanderdonckt@uclouvain.be;

Abstract

Runtime adaptation of graphical user interface elements is a key requirement in the development of intelligent user interfaces. To enable such adaptation, systems must be able to identify user characteristics accurately and in a non-intrusive manner, allowing the construction of user profiles that support more comfortable, safe, and effective interaction. This paper proposes a set of user metrics for collecting interaction data and estimating user characteristics at runtime, grounded on a previously defined user model. Based on a targeted literature review, the metrics are classified into automatic and manual depending on whether their parameters can be captured during interaction. As a main contribution, we define the parameters required to operationalise each metric, enabling their systematic computation. We propose the definition of metrics for 19 UCs, four of which (knowledge, skills, device, and interests) were empirically validated through an experiment involving 60 users. The experiment aimed to evaluate the degree of alignment between the characteristics derived from the proposed metrics and the users' self-reported preferences. It included 15 interactive tasks across two scenarios involving two interactive applications: a home automation system and an autonomous car. The results show significant correlations for all the user characteristics studied, supporting the feasibility of using interaction-based metrics to inform adaptive user interface design. Overall, this work represents a step towards a methodological foundation for designing intelligent user interfaces that dynamically adjust to user behaviour and actions at runtime.

Keywords: Empirical comparison, user characteristics, user metrics, interface customisation.

1 Introduction

Intelligent User Interfaces (IUIs) define a set of rules and behaviours that adapt graphical interface elements to users' diverse characteristics. Achieving this requires an understanding of how users interact with systems and the design of adaptable user interfaces [1]. One of the key challenges in implementing an IUI is determining how to extract User Characteristics (UC). To this end, such systems rely on a set of User Metrics (UM). A user metric is a measure used to analyse UCs during interaction with the system to identify user preferences and expected configurations [2]. The information derived from these UM enables the system to construct a user profile. A user profile, following the definition of Wu et al. [3], is the virtual collection of personal data and configuration settings that allows users to be identified within the system and facilitates the analysis and visualisation of the types of users interacting with it.

Customising the User Interface (UI) to the characteristics of each user profile is highly complex, as it requires developing a distinct interface for each profile. To address this challenge, it is necessary to define a set of distinct user profile groups that cluster users according to shared characteristics, thereby enabling the system to adapt to each group of user profiles [4]. For instance, in an e-commerce context, when a user metric detects that a user is located in a country where transactions are conducted in euros, the system may define different user profile groups based on payment options and display product prices in euros accordingly.

Automated user profiling is a challenging task that aims to infer UCs from interaction data rather than relying solely on manually provided information. By continuously analysing user interactions, these approaches can refine user profiles over time, leading to more accurate and personalised recommendations [5]. To construct a user profile, it is first necessary to define the relevant Uc. To this end, we previously proposed a user model that specifies the set of characteristics required to describe users interacting with a system. The methods proposed in this paper are designed to automatically collect the information required to compute the metrics used to populate this user model, thereby enabling the systematic and dynamic characterisation of users [6].

One of the main contributions of this article is the proposal and validation of a set of metrics for estimating UCs and building a user profile. Following the definition of software measurement proposed by Fenton and Bieman [7], a metric is understood as a quantitative measure derived from user data. User data are collected through interaction logs or questionnaires, while the proposed extraction methods collect and process these data to compute the corresponding metric values. Accordingly, we distinguish between (i) user data, which corresponds to the raw information collected from users, (ii) extraction methods, which define how such data are collected and processed, and (iii) metrics, which are the quantitative measures obtained from the collected data. Our approach starts by analysing prior work on defining metrics for extracting UCs for user profiling. The search for prior work was conducted using a Targeted Literature Review (TLR). The results indicate that existing studies define metrics but do not specify the parameters required to calculate their numerical values. Therefore, the second contribution of our work is the definition of these parameters. Finally, the third contribution is the validation of the proposed metrics with real users. For that, we conducted an experiment involving 60 participants. During the experiment, users

interacted with two scenarios while the system computed the UMs. The resulting metric values were then compared with the users' self-reported preferences. Due to limited variability in several UCs within the participant sample, the empirical validation was limited to Knowledge, Skills, Device, and Interests. The validation results for all these UCs revealed a correlation between user preferences and the computed UM values.

This work is structured as follows. Section 2 reviews previous literature on UM. Section 3 presents the proposed set of metrics. Section 4 describes the experimental procedure. Section 5 presents the results. Section 6 discusses these results. Section 7 exposes the threats to validity. Finally, Section 8 concludes the work and outlines potential future research directions.

2 Related work

The identification of metrics to extract users' characteristics has been widely discussed in the literature. Existing work aims to identify users' characteristics for classifying users accordingly. This information is necessary to establish the different groups of users to which the UI will be adapted. Table A of the supplementary material summarises these works.

Some works follow the *Dynamic Difficulty Adjustment (DDA)* methodology, which employs a set of metrics to identify the user based on their skills (i.e., the number of mistakes made by the user) and provide a system customisation that reduces the difficulty perceived by the user. The *Evidence-Centered Design* framework [8] integrates a DDA methodology into educational games to enhance engagement by using metrics to assess users' knowledge and skills, adapting questions accordingly. A questionnaire evaluates learning, engagement, immersion, difficulty, and self-assessed skills. Results show that 67% of users find the framework easy and enjoyable to use. Araujo et al. [9] evaluate a set of tasks to identify users' skills and goals. Authors present a questionnaire in which users indicate the difficulty they observed, their level of immersion, and whether the system has helped improve their memory. Sarkar et al. [10] present a DDA-based skills model that includes: defining difficulty levels for both the game and tasks; analysing player performance based on collected items, task size, skill magnitude, and completed levels; and dynamically adjusting the difficulty to match the user's skills. The authors validate this model with an experiment with real users. The authors conclude that this model helps users to successfully complete video game levels while maintaining their enjoyment and engagement. The main limitation of these works is that the proposed metrics are not user-centred, with all validation phases relying solely on simulations rather than involving real users.

Regarding *accessibility* context, some studies focus on defining metrics to identify users with disabilities. These works aim to develop customisations that minimise user errors caused by their disabilities. Knisely et al. [11] propose a methodology to group users according to their disability so that their smart devices would adapt to that disability. The methodology draws on data from the US National Health and Nutrition Examination Surveys (NHANES) to map UCs and establish the behaviours of the UI components. Medical specialists evaluated this methodology and revealed that it enabled users with disabilities, whether sensory, physical, perceptual, or cognitive,

to better perform the required tasks. Araim et al. [12] design a system that can automatically capture the different emotions of disabled users while they interact with a system. For this, the system uses a set of neural helmets to read and classify four types of brainwaves to describe these emotions. To validate this system, the authors carry out a validation with real users. This validation concluded that 93% of the analyzed emotions are identified by the system. Pagiatakis et al. [13] design a protocol-driven intervention solution to manage the medication of hypoglycemic users. The system includes sensors, such as a microphone for voice recognition, to identify the user context and determine when a user needs medication. This system was validated with healthcare professionals, and the results show that it can be a first step towards developing a mobile application to help hypoglycemic users. The main limitation of these works is that most of the defined UCs are obtained manually.

Regarding *Recommender systems*, some studies focus on capturing user preferences to provide better recommendations. To achieve this, they incorporate a set of metrics designed to identify user preferences. The Hidden Markov Model (HMM) [14] aims to enhance the recommendations provided by a social media system. To achieve this, the system analyses all actions performed by the user, such as their list of favourite comments and the profiles of users they follow. Then the system provides a list of related content that it considers a match for the user's preferences. The authors validate this model with real users, and the results show that HMM is adequate for collecting and classifying users. The *Personalised Unexpected Recommender System (PURS)* [15] aims to identify unexpected actions performed by the user to enhance recommended systems. By incorporating the user's unpredictability into the recommendation process, the system uses metrics such as video ratings and Click-Through Rate estimates to determine when the user is engaged with the recommendation and when it fails to meet his/her expectations. PURS was validated through an A/B test with a 3% increase in the average video view rate using this model. Jung et al. [16] propose a user profiling method to improve recommendations based on users' preferences and interests. To achieve this, they vector both users and objects within the system, allowing them to determine which object is most appealing to the user. This system was validated using Leave-One-Out Cross-Validation, and the results show that this algorithm improves recommendations when users' interests and preferences differ across user groups. The main limitation of these works is that their findings are specific to an audiovisual context, limiting their applicability to other domains.

Regarding *Adaptive User Interfaces (AUI)*, some studies focus on implementing methods to determine the user's context and provide customisations. *Echat* [17] is a system that implements an AUI that adapts to the user based on his/her current emotional status, detected through facial emotion recognition, to display a user interface that evokes positive feelings to the user, increasing their User Experience (UX). The authors compare the UX of this system with that of a static graphical user interface. They report that most users perceive a better UX with Echat. Khan and khusro. [18] propose an Android application that dynamically adjusts the GUI in real time to minimize driver distractions. The system determines the user's context using sensors such as GPS and velocity sensors while also considering user preferences and device usage to adapt UI components. Evaluation via user questionnaires revealed that 80%

of participants believed the system could reduce distractions and expressed willingness to use it. Awada et al. [19] present an AUI focused on elderly people suffering from dementia. To achieve this, the system combines users' ages and dementia status with an AI algorithm to detect their current emotional state via gesture recognition using a Kinect device. The authors present a questionnaire, and results show that around 30% of the users indicate that this system improves navigation and enhances their UX. The main limitation of these studies is the low statistical data due to a small sample size in the validation phase.

Previous approaches provide different UMs for identifying the user interacting with the system. Existing approaches are specialized for specific contexts. Our work builds on previous findings by combining various concepts to develop a set of UM that can be applied across a wide range of contexts. Using existing metrics from the literature as a foundation, we have defined the parameters required for their calculation, ensuring that the UM can be computed through a clear process.

3 Definition of the Metrics for User Characteristics

In this section, we present one of the key contributions of this work: the definition of UM, designed to capture and quantify UCs.

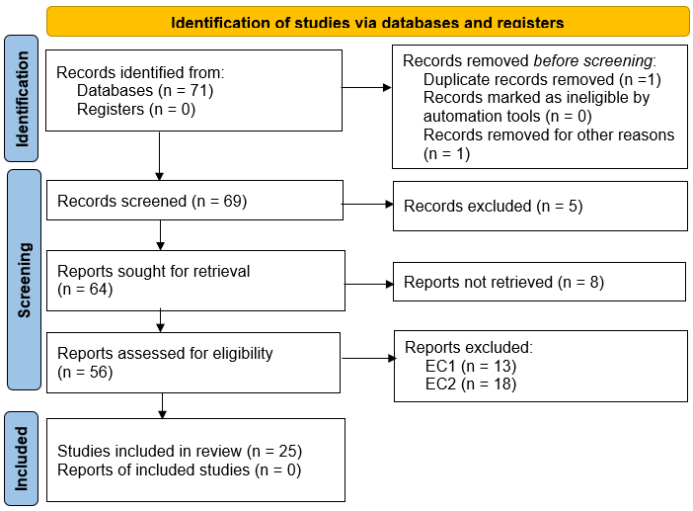


Fig. 1 Included and excluded records of the Targeted literature review.

3.1 Targeted Literature Review

To define the UM, we conducted a TLR [20]. The objective of the review was to identify well-defined, empirically supported user metrics that could be operationalised within our framework, rather than to provide an exhaustive survey of all research on intelligent user interfaces. We searched Scopus and Google Scholar using the Title, Abstract,

and Keywords fields with the query: ("user metrics" AND ("intelligent user interface" OR "adaptive interfaces" OR "smart user interface" OR "model-based UI development" OR "end-user development" OR "personalisation research")) OR ("detect user characteristics" AND "human-computer interaction").

Studies were included if they: (IC1) provided a clear definition and computational formulation of a user metric; (IC2) reported an empirical evaluation involving real users; and (IC3) presented metrics whose definitions were sufficiently general to support application beyond a single implementation context. Studies were excluded if they: (EC1) did not address metrics for extracting or characterising UCs, or (EC2) did not provide sufficient methodological detail to operationalise the proposed metric.

The initial search identified 71 records, of which 2 were removed prior to screening. Sixty-nine records were screened, five were excluded, and 64 reports were sought for retrieval, of which eight could not be obtained. The remaining 56 reports were assessed for eligibility according to the defined criteria, resulting in 31 exclusions (EC1 = 13; EC2 = 18). Ultimately, 25 studies were included in the final review. Figure 1 summarises the different phases of the review process. The extracted metrics were classified into automatic and manual metrics, with priority given to metrics that could be automatically computed within the proposed framework. Now, we provide detailed descriptions of the metrics.

3.2 Automatic metrics

These metrics can automatically capture the following UCs: knowledge, skills, goals, device, language, location, context, and interests. According to previous literature, some of these UCs are derived from one or several metrics.

In the case of the metrics for the UCs (knowledge, skills, and goals), their calculation depends on the values of several parameters. As a contribution to this work, we explicitly define the parameters required to compute these metrics, since previous studies do not specify them. Because each UC is assessed through several heterogeneous metrics, their values are first normalised to a common three-level ordinal scale (1 = low, 2 = medium, 3 = high) before aggregation. Although the scale definition is part of our contribution, the selection was not arbitrary. Instead, they were systematically established by partitioning each metric's domain into three ordinal levels, balancing the possible values across these levels while preserving the metric's semantic meaning [21].

This normalisation makes the metrics comparable and prevents differences in their original scales from influencing the final score, following the recommendations for constructing composite indicators [22]. The overall score of each UC is then computed as the arithmetic mean of the normalised metrics. Since no validated thresholds or weighting schemes exist for these metrics in the literature, the proposed thresholds and equal weighting constitute a methodological design choice introduced in this work to provide a simple, transparent, and reproducible aggregation procedure. This aggregation strategy also mitigates the impact of conflicting metric values, as the final score is determined by the combined evidence from all normalised metrics rather than by any individual observation.

3.2.1 Knowledge

The aim of assessing the user's knowledge is to identify the concepts the user already understands within the system's context or content. The primary challenge with this characteristic is that the user's knowledge is continually evolving. Therefore, the UMs must also detect this knowledge evolution and adapt the UI accordingly [23]. Table 1 presents the user metrics used to determine the user's knowledge and their corresponding values. Additionally, the table includes the source reference used in the TLR to define this metric. The value of the characteristic is calculated as the average of all the metrics into which the characteristic is divided. To ensure a balanced average among the different metrics, the values are normalized into three levels: 1 for low knowledge, 2 for medium knowledge, and 3 for high knowledge. This process is similarly applied to the other characteristics, but we do not repeat it for brevity.

Table 1 User metrics and their parameters for estimating the user's knowledge.

Metric	Definition	Normalization
Mouse clicks (MC) MC=$\lvert u_{clicks} - uc_{es} \rvert$ [24]	Comparison between clicks performed by the user (u_{clicks}) and estimated clicks (uc_{es}).	MC $\leq$ 1; 3 1 $<$ MC $\leq$ 2; 2 MC $>$ 2; 1
Mouse scrolling (MSC) $MSC = \lvert scr_{user} - scr_{es} \rvert$ [23]	Comparison between scroll performed by the user (scr_{user}) and estimated scroll (scr_{es}).	MSC $\leq$ 20; 3 20 $<$ MSC $\leq$ 200; 2 MSC $>$ 200; 1
Search number (SN) $SN = \sum$(clicks on search) [24]	Obtain the number of clicks on the search button.	SN = 1; 3 2 $\leq$ SN $\leq$ 3; 2 SN $\geq$ 4; 1
Accuracy (A) $A = \lvert uc_{user} - u_{misc} \rvert$ [25]	Comparison between user clicks (uc_{user}) and misclicks, clicks outside of available options (u_{misc})	A = 0; 3 0 $<$ A $\leq$ 2; 2 A $>$ 2; 1
Total time (TT) $TT = \lvert time_{user} - time_{es} \rvert$ [26]	Obtain the difference between user time to complete the task ($time_{user}$) and estimated time ($time_{es}$).	TT $\leq$ 10; 3 10 $<$ TT $\leq$ 30; 2 TT $>$ 30; 1
Time per page (TP) $TP = \frac{time_{total}}{pages_{accessed}}$ [27]	Obtain the time per page of the user.	TP $\leq$ 15; 3 15 $<$ TP $\leq$ 30; 2 TP $>$ 30; 1
Sensible pauses (SP) $SP = \sum$(inactivity $> 5s$) [26]	Obtain the sensible pauses (inactivity longer than 5 seconds) made by the user.	SP = 0; 3 0 $<$ SP $\leq$ 3; 2 SP $>$ 3; 1
Complete ratio (COM) COM = Task completed? [26]	Obtain if the user completes the task.	COM $\geq$ .9; 3 .9 $>$ COM $\geq$ .6; 2 COM $<$ .6; 1
Query complexity (QC) Qc = query complexity [23]	Obtain the complexity of the queries formulated by the user. To estimate that, we will follow Ben's classification [23].	QC = 3; 3 QC = 2; 2 QC = 1; 1

After estimating each UM, the system calculates the average of the values obtained to determine the user's knowledge. An average value between 1.00 and 1.66 indicates low knowledge; 1.67 to 2.33, medium knowledge; and values above 2.33, expert knowledge.

3.2.2 Skills

The aim of assessing the *user's skills* is to identify the user's capacity to interact with the system without errors. To achieve this, the system analyses the user's actions and behaviours during interaction to identify instances of errors [28]. The primary challenge with this metric is distinguishing between user errors resulting from limited system knowledge and those stemming from inherent low skill. Table 2 presents the UM used to assess the user's skills, with its parameters and their normalization. Note that the *Accuracy (A)*, *Total Time (TT)*, and *Completion Ratio (COM)* metrics are calculated in the same way as for knowledge characteristics (see Table 1). For example, while longer TT or lower A in the Knowledge characteristic may reflect difficulties in understanding the system or identifying the correct information, in this context, they are interpreted as indicators of procedural proficiency, where more skilled users perform tasks more efficiently and perform fewer execution errors.

After estimating each UM, the system calculates the average of the values obtained to determine the user's skills. An average value between 1.00 and 1.66 indicates that the user has low skill, a value between 1.67 and 2.33 indicates that the user is moderately skilled, and a value above 2.33 indicates that the user is expertly skilled.

Table 2 User metric parameters for estimating the user's skills.

Metric	Definition	Normalization
Click options (CO) $CO = \|cuo - options_{es}\|$ [23]	Comparison between the options accessed by the user (cuo) and the options estimated ($options_{es}$).	CO = 0; 3 $0 < CO \leq 2$; 2 $CO > 2$; 1
Special actions (SPA) $SPA = AVG(\#Sp_{act})$ [28]	Obtain the average of special actions (sp_{act}) performed by the user, including actions such as advancing search.	SPA $> .9$; 3 $.9 \geq$ SPA $\geq .7$; 2 SPA $< .7$; 1
Revisited ratio (R) $R = \|dov - ov\|$ [28]	Comparison between the total time of options (ov) accessed by the user and the different options accessed by the user (dov)	R = 0; 3 $0 < R \leq 3$; 2 $R > 3$; 1

3.2.3 Goals

The aim of assessing the *user's goals* is to identify the user's objectives when interacting with the system. The primary challenge with this metric is identifying all of the user's goals, as users typically have both explicit and implicit goals when interacting with a system. For instance, when a user searches for a t-shirt on an e-commerce platform, their explicit goal is to purchase that specific t-shirt. However, he/she may also have implicit goals, such as considering the t-shirt's brand or the materials used in its production. Therefore, this user metric must also detect these explicit and implicit goals and adapt the UI accordingly [29]. Table 3 presents the UM employed to determine the user's goals with its parameters and their normalisation. Note that *Complete Ratio (COM)*, *Search Number (SN)*, and *Query Complexity (QC)* are calculated as knowledge subcharacteristics (see Table 1). For example, a higher SN or greater QC may result from users deliberately broadening or narrowing their search according to their objectives, rather than reflecting limited knowledge or procedural skills.

Table 3 User metric parameters for estimating the user's goals.

Metric	Definition	Normalization
Options per query (OPQ) $OPQ = \frac{ov}{queries}$ [29]	Obtain the options accessed by query.	OPQ $\leq$ 1; 3 1 $<$ OPQ $\leq$ 3; 2 OPQ $>$ 3; 1
Time max per page (T_{MPAG}) $T_{MPAG} = \max(time\ pages)$ [27]	Obtain the page where the user has spent the most time.	$T_{MPAG} \geq 10$; 3 10$> T_{MPAG} \geq$5; 2 $T_{MPAG} < 5$; 1

After estimating each UM, the system calculates the average to determine the user's goals. An average value between 1.00 and 1.66 classifies the user's goal as informational, while a value between 1.67 and 2.33 classifies it as navigational. A value above 2.33 indicates the user's goal is transactional.

For device, language, location, context, and interests, UM does not depend on parameters; it can be directly obtained without calculation. Next, we define the UM to extract these UCs.

3.2.4 Device

The device characteristics aim to identify the user's preferred method of interaction with the system. Sandhya et al. [30] propose analysing the device's Medium Access Control (MAC) direction to determine its type. Expanding on this approach, we propose analysing screen resources, screen dimensions, and the operating system to classify the user's device. These metrics are particularly relevant because the user's device may influence how they interact with the system. For example, if a user interacts with a mobile device, the graphical elements are smaller than when interacting with a tablet device. The system detects the user's devices, including personal computers, tablets, mobile phones, smart devices, and other similar devices.

3.2.5 Language

The language characteristic aims to determine the most suitable communication channel for the user. We consider this characteristic because user engagement is reduced when users do not understand the GUI content. For instance, if the user only understands Spanish, he/she will refuse to interact with a system written in German. The main limitation in establishing this language is that a user may be proficient in multiple languages. For this reason, the system identifies two different metrics to determine the user's language. The first metric involves identifying the operating system language by its code. The second metric consists of identifying the keyboard's language [31]. When the values obtained by the metrics match, the system identifies that language as the user's default language. If they do not match, the system asks the user to select the most suitable language between those detected.

3.2.6 Location

The location characteristic aims to identify the user's geographical context, including both their current location and, when relevant, their nationality or place of origin. Identifying this characteristic is important for adapting to the user because there is a set of behaviours linked to the user's nationality (user's culture). For this purpose, the system calculates two metrics. The first metric involves identifying the user's current location based on the GPS sensor value. The second metric involves obtaining the user's current location based on their public IP address [32]. Once each metric is calculated, if the values of the two metrics match, the system will indicate this location in the user's profile. In cases where the values obtained by the metrics do not coincide, the system will ask the user which one he/she prefer.

3.2.7 Disabilities

The disabilities characteristic aims to determine if the user has limitations to interact with the system to establish a set of special commands to facilitate his/her interaction. For example, a blind user needs a set of special commands to be able to interact with the system. The main challenge in identifying user disabilities lies in determining what types of special commands the system needs to offer to the user. For this purpose, the system proposes two metrics. Assistive Technology (AT) consists of a set of tools, software, and special devices focused on helping persons with disabilities. The first metric consists of performing a complete analysis of the device to find any AT devices [33]. The second metric involves analysing the user's complete device configuration, particularly accessibility settings, such as screen readers for blind users [34].

3.2.8 Context

The context characteristic aims to define the current situation of the user to provide the most appropriate interaction mode for the user's context. For example, if the user is in a noisy environment, he/she may prefer to interact with the system using touch commands. The main challenge in identifying the user's context lies in providing all the necessary sensors to identify the current context, as it may happen that a user does not have them or does not want to use them. To address this, the proposed metric involves using different sensors on the device. These sensors aim to determine if the user is moving, alone or accompanied, able to communicate with the device through speech or commands, and whether it is outdoors or at home [35].

3.2.9 Interests

The interests characteristic aims to obtain the likes and preferences that a user might have while interacting with the system. For example, a sports enthusiast may seek information related to his/her favorite sports, or a user planning a trip may be interested in travel products such as suitcases. The main challenge in identifying the user's interests arises from the lack of differentiation between an interaction based on the user's interests and one based on some other characteristic. For this purpose, the system combines two metrics. In the first metric, the system analyses the various types

of web pages or system options accessed by the user [35]. In the second metric, the system records whether the user accesses any recommendations it offers [36].

3.2.10 Emotions

The emotions characteristic aims to capture the set of experiences and feelings that the UX has while interacting with the system. For example, when a user is reading a newspaper but becomes frustrated with excessive ads, they may leave the website for a less intrusive alternative. The main challenge in identifying the user's current emotional status lies in the computational complexity required to predict the array of emotions and the difficulty of obtaining the sensors needed to capture them, such as a neural helmet. To capture this, the system implements a metric that analyses user movements, facial expressions, and vocal cues to assess their emotional state [37].

Table 4 Metrics to estimate the manual characteristics.

UC	Purpose	Challenge	Metric
Age [38]	Adapt system responses according to the user's age range.	Is personal data, and users may be reluctant to provide it.	Allow users to select an age range during registration.
Gender [38]	Identify trends and behaviours expected by the user.	Users may search for products on behalf of another person.	Enable labels to specify the gender for which they wish to search for products.
Job characteristics [39]	Estimate users' expertise to adapt response complexity based on their professional background.	Difficult to infer whether an interaction is for personal or professional purposes.	Include optional fields in the registration form to specify job role, professional profile, and experience level.
Study Level [40]	Adapt the technical depth of the information to the user's knowledge background.	Knowledge may also come from informal learning rather than formal studies.	Including in the registration form an option for users to indicate their current study level.
Preferences [41]	Customise system visuals, notifications, and functionalities according to user-selected options.	Subjective and variable over time; preferences may change in future interactions.	The system should include a dedicated page for user preferences, where users can input and modify them as needed.
Income [42]	Estimate purchasing capability to customise product or subscription suggestions.	Users may have different budgets depending on the product or search context.	Creating various income groups, allowing users to select the group that best represents their income level.
Hobbies [43]	Identify users' leisure activities to support personalised recommendations.	High variability makes it difficult to infer.	Allow users to enter and update their hobbies at their convenience.

3.3 Manual metrics

Despite efforts to automatically calculate UC values, some characteristics depend on user-entered information through various forms. Users may enter the required data for manual metrics incorrectly, or they may be unwilling to provide it. The primary challenge in defining these metrics is presenting them to users in a way that ensures they understand and are willing to input them, without compromising their engagement

or the system's usability. In this work, we define the following metrics to be manually obtained: age, gender, job, job profile, job experience, study level, preferences, income, and hobbies. Table 4 summarises these characteristics.

4 Experimental Validation

The experiment is situated within the context of Human-Computer Interaction (HCI). Its objective is to compare the characteristics derived from the proposed metrics with users' self-reported preferences. Among the metrics defined in Section 3, the evaluation focuses on the automatic metrics, since manual metrics are directly provided by users and therefore do not allow the analysis of correlations between metric values and user preferences. The analysis is based on the extraction of metrics during user interaction with the system. The study is conducted from the perspective of researchers and practitioners interested in understanding how UCs can be inferred from interaction data and applied to the design of user interfaces.

4.1 Research Questions and Hypothesis Formulation

Next, we define the research question for the experiment, which aims to assess the effectiveness of the metrics for extracting UCs.

- *RQ1*: Is there a correlation between the value of the characteristics calculated with the UM and the value estimated by the user according to their preferences?. And the null hypothesis H_{01}: There is no correlation between the values obtained using the metrics and those estimated by the user based on their preferences.

4.2 Participants

Participants were recruited from the Computer Engineering degree program at the University of Valencia, Spain, during the 2023-2024 academic year. Although respondents were end users and did not need prior experience in real-world software project development, they underwent a selection process to assess their knowledge of interface programming and software modelling. We selected 60 participants (9 female, 51 male) (M= 20.30, SD= 1.73, Mdn = 20). All participants signed a GDPR-compliant consent form along with a demographic questionnaire. All participants were students with little experience working in computer science roles. (M= 0.36, SD= .63, Mdn=0).

4.3 Apparatus and Setup

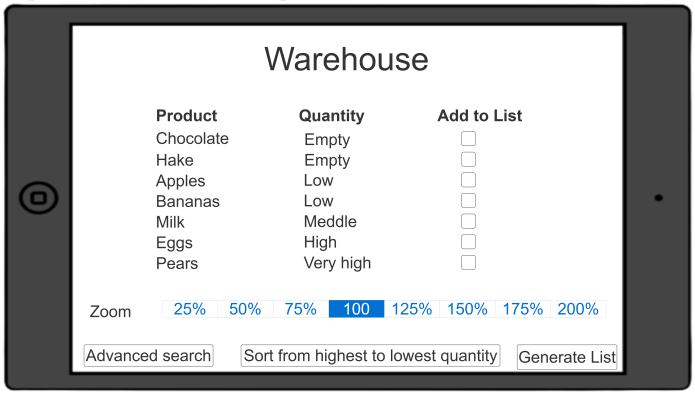
The experiment involved a single device through which the user interacted, along with a set of prototypes simulating the UIs of two systems: a home automation system and an autonomous car system. Next, we present examples of the UI for each scenario.

Figure 2 represents the UI for the warehouse option within the home automation system. Here, users can view the warehouse's current status, add products to a shopping list for purchase at a supermarket, adjust the page's zoom level, apply filters, and perform advanced product searches.

In this experiment, participants were exposed to QuantUX running on their computers to obtain the parameters needed to estimate their characteristics and determine

their preferences. Initially, participants completed a form to record their demographic characteristics (age, gender, education level, among others), to obtain an accurate profile of each individual. Subsequently, they were asked to interact with a series of user prototypes designed on the platform, where they were required to perform specific tasks or navigate through the proposed interfaces. During the interaction, metrics for measuring UCs were automatically calculated in the background, remaining hidden from the user's perspective.

Fig. 2 An example of a UI used in the experiment.



After interacting with the prototypes, users filled out a second form, where they expressed their users' preferences. This enabled us to compare their preferences with the values calculated using our proposed metrics.

4.4 Independent and Dependent Variables

The experiment uses a within-subject design with one independent variable (factor) and eight dependent variables (response variables). As an independent variable, we have *Procedure to calculate UCs*, which represents the method used to extract the UC. This independent variable has two levels: one for the value of the UC calculated using the metrics proposed in the article, and the other for the users' self-reported preferences. This allows us to analyse, for each UC, the correlation between the value extracted using the proposed metrics and the value provided by the users themselves. Regarding dependent variables, we have one variable per UC calculated using automated metrics: *Knowledge*, *Skills*, *Goals*, *Device*, *Language*, *Nationality*, *Context*, and *Interests*. Table B of supplementary material [1] exposes the dependent variables and their definition. Note that *Disabilities* were not included as dependent variables due to the complexity of their assessment.

[1] https://doi.org/10.5281/zenodo.17736074

4.5 Tasks and Scenarios

The experiment comprises two scenarios of similar complexity, each containing straightforward tasks designed to limit the duration to one hour. Each scenario includes multiple tasks, with each task implementing a specific proposed metric to calculate UCs for evaluation. We ensure that the metrics are balanced among the number of tasks. We specify the metric and characteristic evaluated in each task in the task section of the supplementary material. Next, we define each scenario:

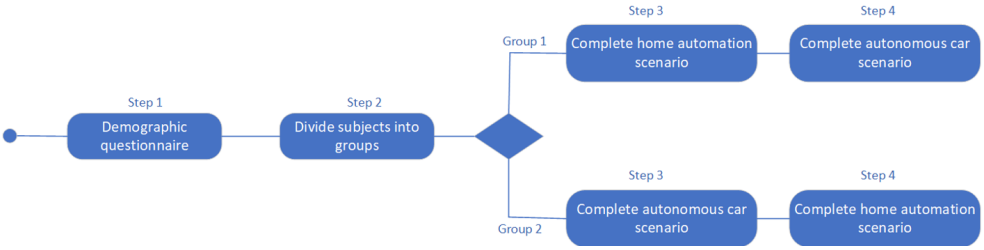
- *Experimental scenario 1. Home automation system.* This scenario aims to design the UI of a home automation system.
- *Experimental scenario 2. Autonomous car.* This scenario aims to design the UI of an autonomous car system.

4.6 Procedure

The experiment consisted of a single session in which a set of prototypes was presented to the subjects to complete the tasks in the two experimental scenarios. The entire process was guided using a website that included the tasks and questionnaires. To accomplish this experimental process, we defined four steps, as depicted in Figure 3. The steps are described as follows:

- First step. Complete a demographic questionnaire.
- Second step. Divide subjects into 2 groups to complete the experiment. Subjects within Group 1 have the order: Scenario 1, Scenario 2. Subjects within Group 2 have the order: Scenario 2, Scenario 1.
- Third step. Complete the tasks in the first experimental scenario while the system captures the specific UC for each task. After each task, the website guiding the procedure prompts the subject to specify his/her preferred value for the previously calculated UC, enabling a comparison between both values.
- Fourth step. Complete the tasks of the second scenario as the third step.

Fig. 3 Graphical representation of the steps of the experiment.



4.7 Metrics and data collection

Data collection was conducted during the task execution phase, in Steps 3 and 4 of the experimental procedure. The process was different depending on whether the UCs

were calculated using our proposed metrics or whether they were specified based on users' preferences (the two levels of the independent variable):

1. Data collection from the metrics proposed in Section 3: While the subject performed the experimental tasks (defined in supplementary material), the proposed metrics were applied. The parameters of each metric were assigned as the user interacted with the system. Depending on the task, the calculated metrics corresponded to one or several UCs. Once each task was completed, the system automatically normalized each metric and then calculated the average by gathering the metrics associated with the same UC, which represented the value of that characteristic. This value was stored for comparison with the user's preferences.
2. Data collection from users' preferences: Once each experimental task was completed, the system explicitly asked the user for their preferences regarding the UCs involved in the task. Based on the user's experience with the task, the subjects evaluated the UCs within the same range of values as the metric being evaluated.

4.8 Data analysis

We have conducted two different types of statistical tests. First, we analysed whether demographic data can influence users' characteristics, both when calculated using our proposed metrics and when specified by the users' preferences. Second, we examined the possibility of a linear relationship between the UC values calculated using our proposed metrics and those extracted from users' preferences. The more precise our metrics are, the greater the correlation should be.

4.8.1 Study of demographic data

The statistical analysis of the demographic data was conducted using a mixed-model statistical test [44], which combines a repeated variable and a between-subjects variable. The repeated variable, in this case, is the experimental scenario. All subjects performed the experimental tasks in both scenarios (home automation system and autonomous car). The between-subjects variable, in this case, is the demographic data (age, gender, disabilities, education level, job, job experience, and job profile). The assumption for applying the mixed model is that residuals are normally distributed. The normality of residuals can be tested using the Shapiro-Wilk test, applied to the residuals automatically calculated during the application of the mixed model test [45]. The p-value of the mixed model indicates whether there is a significant difference between the demographic data and the UCs, whether calculated using the proposed metrics or specified by the users' preferences. For variables with significant differences among treatments, we use Cohen's d test [46] to calculate the effect size, which provides the magnitude of the difference. The meaning of the effect size is as follows: values greater than .8 indicate a large effect; values from .79 to .5 indicate a moderate effect; values from .49 to .2 indicate a small effect. Note that Cohen's d cannot be calculated for the interaction between demographic data and scenario.

4.8.2 Study of correlation between levels of the independent variable

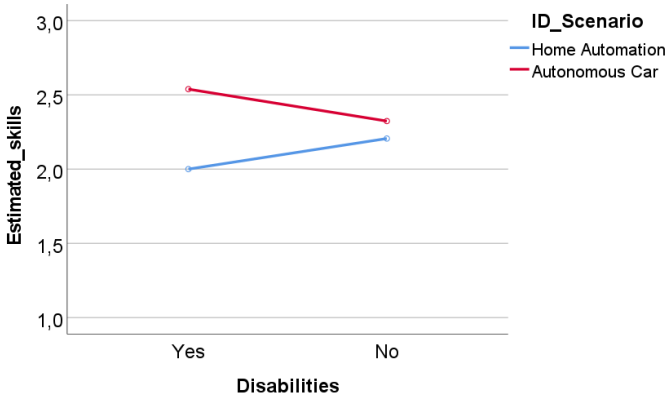
The most suitable statistical test for addressing RQ1 is linear regression. Linear regression is a statistical modelling technique that is used to obtain a relationship between an independent variable and a dependent variable. Through linear regression, we aim to analyse whether there is a correlation between the UCs calculated using the proposed metrics and the user's preferences for adapting the UI. Therefore, we conduct a regression analysis for each UC. When the p-value is lower than .05, we can conclude that there is a statistically significant correlation between the independent variable and the dependent variable [47].

We have ensured that the assumptions of linear regression are met: (1) Linearity; (2) Independence; (3) Homoscedasticity; (4) Normality; (5) No multicollinearity (the independent variables exhibit low correlation among themselves); (6) No endogeneity (there is no association between the errors and the independent variables).

We have also computed the effect size for each independent variable, which quantifies the magnitude of the difference between the groups. This measure is determined using R-squared (R^2) [47], representing the proportion of variance in the dependent variable explained by the independent variable and ranging from 0 to 1. R^2 can be interpreted as follows: a large effect when the value is above .5, a moderate effect when the value is around .3 and .5, and a low effect when the value is below .3.

Statistical power determines whether we have a sufficient sample size to reject the null hypothesis and identify significant differences. According to G*Power [48], for a moderate effect size (.5) in a within-subjects design, we need a sample size of over 34 subjects to achieve good power values (.96). In this experiment, we have a sample of 60 subjects, which exceeds the minimum required size to achieve .96 of power.

Fig. 4 Correlation between estimated skills and user disabilities across different scenarios.



5 Results

Next, we describe the mixed model used to analyse whether demographic data (age, gender, disability, education level, job, job experience, and job profile) significantly

influence participants' UCs, as calculated using automated metrics. We also analyse possible interactions between these demographic data and the user's characteristics. The only significant demographic characteristic is job experience with the system's estimated skills. In this case, Cohen's d is .498, indicating a moderate effect size. This significant result can be attributed to the lack of heterogeneity in the data, as there is only 1 subject from vocational training and 59 subjects from high school. Regarding the interactions between factors and scenarios, we find significant results for age, disability, study level, and job experience. The results of the mixed model are presented in Tables C and D of the supplementary material [2]. As an example of a plot used to analyse the significance of these interactions, Figure 4 shows the profile plot used to analyse the significant interaction between Disabilities and the experimental problem for Estimated Skills. The plot shows that the Autonomous Car scenario yields higher Estimated Skills for participants with disabilities. This may be due to their greater familiarity with car display systems than with smart home systems, which could lead to differences in perceived skills.

All significant interactions between factors and scenarios can be justified by the unbalanced demographic data. For gender, there are 9 women and 51 men; for job, most subjects are students with prior computer science experience; and for age, most students are 20 years old.

5.1 Linear regression

In this section, we present the results of the linear regression analysis examining the potential correlation between UC values derived from automatic metrics and users' preferences. This linear regression analysis aims to determine whether the system's automatic UM classification aligns with user preferences. For UCs showing a significant correlation between the calculated value and the metrics, as well as the user's preferences, we include a plot to aid interpretation of these relationships. Table E in the supplementary material summarises each user metric, including its *p-value*, effect size (R^2), and the corresponding regression formula. Figures A, B, C, and D of the supplementary material [3] present the linear regression for each evaluated characteristic. From all the UCs calculated using automatic metrics, goals, language, location, and context cannot be calculated because they are too homogeneous among the recruited subjects. This lack of variability prevents the independent variable from being suitable for creating a regression model, and thus, no *p-value* is provided for these UCs.

Results indicate that all UCs analysed *Knowledge*, *Skills*, *Device*, and *Interests* exhibit a statistically significant correlation with the dependent variable (*p-value* $<$ .001). However, the effect size varies considerably among them. *Device* shows the strongest correlation, with an R^2 of .95. However, as the experimental setup included only three device categories and device type was directly observable, this result is best interpreted as a ceiling effect under constrained conditions rather than evidence of the UM's discriminative capability. Consequently, it should not be generalised to more heterogeneous scenarios. In contrast, *Knowledge*, *Skills*, and *Interests* present

[2] https://doi.org/10.5281/zenodo.17736074
[3] https://doi.org/10.5281/zenodo.17736074

low effect sizes (R^2 = .21, .18, and .12, respectively), indicating that a significant portion of the variance remains unexplained, likely due to additional influencing factors not included in the model. Although these relationships are statistically significant, they account for less than one-quarter of the variance in user self-reports, indicating limited predictive power and suggesting that additional factors not captured by the proposed metrics contribute to these UCs. We additionally computed Spearman's rank correlation coefficients, which are more appropriate for ordinal data. The coefficients obtained for *Knowledge*, *Skills*, *Interests*, and Device were .429, .395, .426, and .984, respectively. These results are consistent with the regression analysis, confirming the very strong association for *Device* and the moderate associations for the remaining UCs. Despite these differences, all characteristics display an upward linear trend, reflecting a positive relationship with the dependent variable. The variability in dispersion across characteristics highlights the differing levels of predictability offered by each feature, with some requiring further investigation to identify complementary explanatory variables.

So, we reject H_{01}, which states that there is no correlation between UCs extracted with automatic metrics and user preferences. We can only demonstrate that the correlation exists for skills, knowledge, device and interests UCs.

6 Discussion

This section discusses the results by comparing them with previous evaluations and the experimenters' expectations. When the results differ from the expected outcomes, possible explanations are analysed. The discussion is organised by UC.

Regarding user *skills*, there is a significant correlation between the user's skills value as measured by the metric and their preferences. Reguera-Bakhache et al. [49] state that by reducing the total time and the number of interactions for advanced users, the system achieves 86.75% accuracy in classifying user types. This idea aligns with our results; in our case, if customisation reduces the total time (TT) (see Table 1) and options accessed by the user (CO) (see Table 2), the correlation between the metric values and user preferences becomes stronger. This means that users who required less time and made fewer clicks were correctly classified by the system based on their skill preferences. On the other hand, the special action (SPA) metric (see Table 2) has been the most complex to implement and introduces the most uncertainty in estimating user skills. This is because the prototypes did not support all possible special actions, such as shortcuts or advanced searches, which are not available in the test.

Regarding user *knowledge*, there is a significant correlation between the value obtained with the metric and the users' preferences. Carrera-Rivera et al. [50] argue that adaptive user interfaces should rely on user models that accurately capture relevant UCs, together with contextual information and user interactions, to provide personalised interfaces and improve usability and UX. In our experiment, however, only 62.50% of users were correctly classified based on their knowledge preferences. The system overestimated users' knowledge in 22.50% of the cases and underestimated it in 15.00%. These misclassifications may be explained by the fact that the Accuracy

(A) and Query Complexity (QC) metrics (see Table 1) are influenced by the experimental protocol, since the predefined instructions specified the interactions and queries participants had to perform. This task dependency is inherent to these behavioural metrics and does not invalidate their use to estimate Ucs within a given interaction context, as both the objective measurements and the users' subjective preferences were elicited under the same experimental conditions. Rather, it limits the direct generalisation of the observed relationships to different tasks, which should be investigated in future work.

Users perceive an improved experience and greater usability when the system adapts to the *device* they are using. By adjusting the graphical user interface to ensure all components remain fully visible and interactive, the system enhances usability. This adaptation includes optimising size, colour, and layout to reduce user errors [51]. These expectations are met because subjects were able to complete the experiment on their devices. However, the experimental context included only three device categories, which likely produced a ceiling effect in the Device estimation results. Therefore, this finding should be interpreted as valid within the experimental conditions rather than as evidence of the framework's general performance.

Regarding *interests*, there is a significant correlation between the metric values and users' preferences. Bedi et al. [52] state that user interests are the most difficult and complex characteristic to adapt to, given the wide range of personal interests and the lack of clear relationships between user interests and other UCs. This is not reflected in the experiment because the application contexts were predefined, leading to a high correlation between user interests and preferences.

Furthermore, within our validation, we observed several relationships among the UCs. One of these relationships is that 55% of users classified as intermediate-skilled are also classified as middle-knowledge users, indicating that about half of users consider themselves 'standard' users of such systems. Another relationship is that 24% of users with visual problems or disabilities are classified as intermediate-skilled users, and 23% of these users are considered to have a high level of knowledge. Additionally, users with visual problems or disabilities tend to have a better estimate from the system based on their knowledge than on their skills. This may be because the system can detect multiple clicks from the user, which are due to difficulties in visualising the icons on the prototypes rather than mistakes. Finally, 78.33% of users with disabilities accessed the system on a laptop, likely because of its larger screen compared to mobile or smart devices.

The experimental validation uses self-reported preferences as the reference for evaluating the proposed metrics. Although self-report is widely used to assess subjective UCs, it should not be interpreted as an objective ground truth. Users may overestimate or underestimate characteristics such as knowledge or skills; therefore, agreement between estimated and self-reported values does not necessarily imply that the proposed metrics capture users' actual capabilities. Instead, the results indicate the degree of consistency between the automatically estimated UCs and users' self-perceptions. This choice is also aligned with the objective of the proposed framework. Since the estimated UCs are intended to drive interface adaptation, users' perceptions are highly relevant. For instance, users who perceive themselves as inexperienced or unskilled

may require a simpler interface regardless of their objectively measured competence. Likewise, characteristics such as preferred devices or personal interests are inherently subjective and are best represented by the users themselves.

In conclusion, we can state that there is a correlation between the UM values and users' preferences. Based on our results, we can derive some recommendations for extracting UC using automatic metrics: (1) It is advisable to correctly identify the set of specific actions that the user can perform when interacting with the system; (2) User interest is the most challenging characteristic to estimate, while the user device is the best-ranked UC; (3) We consider it important to define the flow of steps that the user must follow to complete the task accurately, as the user's skill also depends on how well these steps are executed.

Table 5 Threats to validity of the experiment.

Threat	**Description**	**How we deal with?**
	Conclusion Validity	
Knowledge contamination[1]	Carryover effect between experimental problems.	We divide the subjects into two groups, each starting with a different scenario.
High homogeneity[1]	Subjects too homogeneous	We recruit subjects with similar profiles.
Reduced statistical power[2]	Sample size may be insufficient to detect patterns.	We verify using the G*Power tool that 60 subjects are enough.
Experimental expectations[1]	Experimenters may unintentionally influence results.	We ask the subjects questions about their preferences regarding the UI.
	Internal Validity	
Instrumentation[3]	Experimental tools may impact outcomes.	We conduct a pilot study.
Subject experience[1]	Users' prior knowledge may affect responses.	We design simple tasks that don't require prior experience.
Selection[3]	Natural variations in human actions may affect results.	We recruit all subjects voluntarily and obtain consent.
Interaction with selection[1]	Subjects' behaviour may differ between groups.	We ensure both groups perform the same tasks.
	Construct Validity	
Hypothesis guessing[1]	Subjects may infer the experiment's purpose.	We conceal the research question from users.
Evaluation apprehension[1]	Users may feel nervous about being assessed.	We recruit volunteers and offer an extra class score.
Unclear construct explanation[2]	Constructs may not be clearly defined before measurement.	We define metrics based on TLR results.
	External Validity	
Selection bias[1]	Subjects may not represent final users.	We align user profiles with potential GUI users.
Problem dependency[1]	Results may depend on experimental problems.	We use two contexts: a smart home and an autonomous car.
Context Interaction [1]	Experimental materials may not be realistic.	We include visual prototypes in task questionnaires.

7 Threats to Validity

This subsection argues the threats to validity of our experiment. To identify these threats, we adopt Wohlin's classification [53]: conclusion validity, internal validity, construct validity, and external validity. For each group, we identify the threats that cannot be addressed, those that we could mitigate, and those that we have resolved. Table 5 represents the summary of the threats suffered in this experiment, threats marked as 1 are mitigated, as 2 are avoided, and as 3 are addressed.

8 Conclusion and Future Work

In this paper, we propose a set of user metrics (UMs), classified as automatic and manual metrics, for estimating 19 UCs (job characteristics include job, job experience and work profile UC). Since previous work did not define the parameters required to estimate these metrics, this paper specifies the parameters needed to calculate each UM. Of the 19 UCs, this study evaluates four (Knowledge, Skills, Device, and Interests) by comparing the characteristics estimated using the proposed metrics with the users' self-reported preferences. The evaluation was conducted through an empirical experiment involving 60 participants who interacted with software prototypes. The results showed significant correlations for all four UCs analysed. The remaining 15 UCs could not be empirically evaluated because of the homogeneity of the participant sample and the large variability required across the different characteristics. It should be noted that evaluating characteristics such as Age, Language, or Disability would require sufficiently large and diverse participant groups to ensure adequate representation of the different values for each characteristic. Such a sample size would be impractical to obtain in a single replication of the experiment.

Our approach has some limitations: (1) Some UM rely on users' estimations calculated by an expert in graphical user interface design. For instance, certain metrics are contingent upon the number of clicks, thereby introducing subjectivity into the evaluation process as they hinge on the expert's discretion; (2) Metrics are tailored to a specific user model, intended to be applicable across various contexts. Nevertheless, our approach also boasts several advantages: (1) Metrics are abstract enough to seamlessly integrate into any software development methodology. The ultimate goal of our approach is to ensure that the user metrics can be employed across different application contexts; (2) Metrics are defined with clarity and precision, yet maintain a level of generality suitable for application across diverse software development methodologies; (3) UCs remain pertinent across multiple contexts, and metrics are sufficiently adaptable for specific platforms and interaction modalities; (4) Despite the sample size being relatively homogeneous and the statistical power not being extensive, results demonstrate a correlation, indicating that metrics align with users' preferences.

As future work, we plan to assess the usability of the proposed metrics using the UEQ+ or System Usability Scale questionnaires. Our aim is to determine whether the proposed UM improve the usability of the systems. Additionally, we plan to replicate the validation process with a more heterogeneous group of subjects to ensure the robustness and generalizability of our findings with other demographic data. We also

plan to conduct an end-to-end evaluation that connects metric estimation to adaptation outcomes. Finally, we plan to repeat the experiment with a larger and more diverse participant sample.

Declarations

Funding. This work was supported by the Generalitat Valenciana with TENTACLE under project CIAICO/2023/089, and with the Spanish Ministry of Science with FEDER under the projects PHYLOVAR (PID2024-162114OB-I00) and EU4IRI (PID2024-161104OB-C22).

Data availability. Supplementary Figures A to D, Supplementary Tables A to E, as well as the data obtained from the questionnaire supporting the analysis of results, are available at Zenodo.

Author contributions. All authors contributed to the development of the research, evaluation, and writing of the article.

Conflict of interest. No Conflict of interest.

Ethical approval. Not applicable.

Consent for publication. All the authors have approved the publication of this work.

References

[1] Schmidt, A., Mayer, S., Buschek, D.: Introduction to intelligent user interfaces. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–4 (2021). https://doi.org/10.1145/3411763.3445021

[2] Hou, W., Liu, J., Yan, X.: A questionnaire data clustering method based on optimized k-modes algorithm. In: Artificial Intelligence in HCI - Second International Conference. Lecture Notes in Computer Science, vol. 12797, pp. 326–342. Springer, Virtual Event, July 24-29 (2021). https://doi.org/10.1007/978-3-030-77772-2_22

[3] Wu, D., Du, X., Peng, F.: Multi-layer and multi-source features stacking ensemble learning for user profile. Electric Power Systems Research **229**, 110128 (2024) https://doi.org/10.1016/j.epsr.2024.110128

[4] Abri, S., Abri, R., Çetin, S.: Group-based personalization using topical user profile. In: Adjunct Proceedings of the 28th ACM Conference on User Modeling, Adaptation And Personalization (UMAP 2020), pp. 181–186 (2020). https://doi.org/10.1145/3386392.3399559

[5] Hu, S., Kumar, A., Al-Turjman, F., Gupta, S., Seth, S., Shubham, S.: Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation. IEEE Access **8**, 26172–26189 (2020) https://doi.org/10.1109/ACCESS.2020.2971087

[6] Gaspar, A., Gil, M., Panach, J., Romero, V.: Towards a general user model to develop intelligent user interfaces. Multimedia Tools and Applications, 1–34 (2024) https://doi.org/10.1007/s11042-024-18240-w

[7] Fenton, N., Bieman, J.: Software Metrics: A Rigorous and Practical Approach, 3rd edn. CRC Press, Boca Raton, FL, USA (2014). https://doi.org/10.1201/b17461

[8] Shena, B.A., Sitohang, B., Rukmono, S.: Application of dynamic difficulty adjustment on evidence-centered design framework for game based learning. In: Proceedings of IEEE International Conference on Data and Software Engineering, pp. 1–6 (2019). https://doi.org/10.1109/ICoDSE48700.2019.9092725

[9] Araujo, V., Mendez, D., Gonzalez, A.: A novel approach to working memory training based on robotics and ai. Information **10** (2019) https://doi.org/10.3390/info10110350

[10] Sarkar, A., Cooper, S.: Using a disjoint skill model for game and task difficulty in human computation games. In: Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHIPlay 2019), Companion Extended Abstracts, pp. 661–669 (2019). https://doi.org/10.1145/3341215.3356310

[11] Knisely, B., Vaughn-Cooke, M., Wagner, L., Fink, J.: Device personalization for heterogeneous populations: leveraging physician expertise and national population data to identify medical device patient user groups. User Modeling and User-Adapted Interaction **31**(5), 979–1025 (2021) https://doi.org/10.1007/s11257-021-09305-8

[12] Araim, Z., Khalaf, S., Alheeti, K.A.: A new detection control system for disabled people's wheelchair. International Journal of Advanced Science And Technology **29**(4), 7951–7962 (2020) https://doi.org/10.30569/30086

[13] Pagiatakis, C., Rivest-Hénault, D., Roy, D., Thibault, F., Jiang, D.: Intelligent interaction interface for medical emergencies: Application to mobile hypoglycemia management. Smart Health **15**, 100091 (2020) https://doi.org/10.1016/j.smhl.2019.100091

[14] Elbaghazaoui, B., Amnai, M., Fakhri, Y.: Predicting user behavior using data profiling and hidden markov model. International Journal of Electrical and Computer Engineering **13**(5), 5444–5453 (2023) https://doi.org/10.11591/ijece.v13i5.pp5444-5453

[15] Li, P., Que, M., Jiang, Z., Hu, Y., Tuzhilin, A.: Purs: Personalized unexpected recommender system for improving user satisfaction. In: Proceedings of the 14th ACM Conference On Recommender Systems (RecSys 2020), pp. 279–288 (2020). https://doi.org/10.1145/3383313.3412238

[16] Jung, H., Park, H., Lee, K.: Enhancing recommender systems with semantic

user profiling through frequent subgraph mining on knowledge graphs. Applied Sciences **13**(18) (2023) https://doi.org/10.3390/app131810041

[17] Kumaran, R., Doshi, V., Chen, S., Nargund, A., Höllerer, T., Sra, M.: Echat: An emotion-aware adaptive ui for a messaging app. In: Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST 2023), pp. 1–3 (2023). https://doi.org/10.1145/3586182.361669

[18] Khan, I., Khusro, S.: Towards the design of context-aware adaptive user interfaces to minimize drivers' distractions. Mobile Information Systems **2020** (2020) https://doi.org/10.1155/2020/8858886

[19] Awada, I., Mocanu, I., Nastac, D., Benta, D., Radu, S.: Adaptive user interface for healthcare application for people with dementia. In: Proceedings of the 17th Conference on Networking in Education and Research (RoEduNet 2018), pp. 1–5 (2018). https://doi.org/10.1109/ROEDUNET.2018.8514150

[20] Mangat, G., Sharma, S.: How does the use of different targeted literature review (tlr) methodologies impact the research output? Value in Health **26**(6) (2024) https://doi.org/10.1016/j.jval.2023.03.1572

[21] Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E.: Multivariate data analysis (2019)

[22] OECD, European Union, EC-JRC: Handbook on Constructing Composite Indicators: Methodology and User Guide. OECD Publishing, Paris (2008). https://doi.org/10.1787/9789264043466-en

[23] Ben, W., Jiqun, L.: Characterizing and early predicting user performance for adaptive search path recommendation. In: Proceedings of the Association for Information Science and Technology, vol. 60, pp. 408–420 (2023). https://doi.org/10.1002/pra2.799

[24] Zhou, J., Zahiri, S.M., Hughes, S., Jadda, K.A., Kallumadi, S., Agichtein, E.: Debiased modeling of search click behavior with reinforcement learning. In: The 44th International Conference on Research and Development in Information Retrieval, pp. 1637–1641. ACM, Canada (2021). https://doi.org/10.1145/3404835.3463228

[25] Yu, R., Gadiraju, U., Holtz, P., Rokicki, M., Kemkes, P., Dietze, S.: Predicting user knowledge gain in informational search sessions. In: Proceedings of the 41st International ACM Conference on Research and Development in Information Retrieval, pp. 75–84 (2018). https://doi.org/10.1145/3209978.3210064

[26] Rao, N., Bansal, C., Mukherjee, S., Maddila, C.S.: Product insights: Analyzing product intents in web search. In: CIKM '20:, pp. 2189–2192. ACM, Ireland (2020). https://doi.org/10.1145/3340531.3412090

[27] Yu, R., Tang, R., Rokicki, M., Gadiraju, U., Dietze, S.: Topic-independent modeling of user knowledge in informational search sessions. Information Retrieval Journal **24**(3), 240–268 (2021) https://doi.org/10.1007/s10791-021-09391-7

[28] Rasch, J., Middelbeck, D.: Knowledge state networks for effective skill assessment in atomic learning. CoRR **abs/2105.07733** (2021) https://doi.org/10.48550/arXiv.2105.07733 2105.07733

[29] Berkovitch, O., Caduri, S., Kahlon, N., Efros, A., Caciularu, A., Dagan, I.: Identifying user goals from UI trajectories. In: Companion Proceedings of the ACM on Web Conference 2025, pp. 2381–2390. ACM, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025 (2025). https://doi.org/10.1145/3701716.3717525

[30] Aneja, S., Aneja, N., Islam, M.S.: Iot device fingerprint using deep learning. CoRR **abs/1902.01926** (2019) https://doi.org/10.48550/arXiv.1902.01926 1902.01926

[31] Gothe, S.V., Ghosh, S., Mani, S., Guggilla, B., Agarwal, A., Sanchi, C.: Language detection engine for multilingual texting on mobile devices. CoRR **abs/2101.03963** (2021) https://doi.org/10.48550/arXiv.2101.03963 2101.03963

[32] Mishra, V., Laperdrix, P., Vastel, A., Rudametkin, W., Rouvoy, R., Lopatka, M.: Don't count me out: On the relevance of ip address in the tracking ecosystem. In: Proceedings of The Web Conference 2020, pp. 808–815. Association for Computing Machinery, New York, USA (2020). https://doi.org/10.1145/3366423.3380161

[33] McNicholl, A., Casey, H., Desmond, D., Gallagher, P.: The impact of assistive technology use for students with disabilities in higher education: a systematic review. Disability and Rehabilitation: Assistive Technology **16**(2), 130–143 (2021) https://doi.org/10.1080/17483107.2019.1642395

[34] Šumak, B., Kous, K., Martínez-Normand, L., Pekša, J., Pušnik, M.: Identification of challenges and best practices for including users with disabilities in user-based testing. Applied Sciences **13**(9) (2023) https://doi.org/10.3390/app13095498

[35] Hoi, N.T., Que, T.D.: Discovering user interest in social media based on correlation. Lecture Notes in Networks and Systems **848**, 403–413 (2024) https://doi.org/10.1007/978-3-031-50818-9_43

[36] Gupta, A., Dixit, A.: A novel user trend-based priority assigner and url scheduler for dynamic incremental crawling. Concurrency and Computation: Practice and Experience **34**(3) (2022) https://doi.org/10.1002/cpe.6555

[37] Machová, K., Szabóova, M., Paralič, J., Mičko, J.: Detection of emotion by text analysis using machine learning. Frontiers in Psychology **14** (2023) https://doi.org/10.3389/fpsyg.2023.1190326

[38] Sun, B., Mao, H., Yin, C.: Male and female users' differences in online technology community based on text mining. Frontiers in Psychology **11** (2020) https://doi.org/10.3389/fpsyg.2020.00806

[39] Seveso, A., Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M.: Skills2job: A recommender system that encodes job offer embeddings on graph databases. In: 25th Conference on Artificial Intelligence, 33th Conference on Innovative Applications of Artificial Intelligence, vol. 12605, pp. 15885–15886. AAAI Press, Virtual event (2021). https://doi.org/10.1609/aaai.v35i18.17939

[40] Tondello, G.F., Nacke, L.E.: Validation of user preferences and effects of personalized gamification on task performance. Frontiers Comput. Sci. **2**, 29 (2020) https://doi.org/10.3389/fcomp.2020.00029

[41] Cui, Z., Xu, X., Xue, F., Cai, X., Cao, Y., Zhang, W., Chen, J.: Personalized recommendation system based on collaborative filtering for iot scenarios. IEEE Trans. Serv. Comput. **13**(4), 685–695 (2020) https://doi.org/10.1109/TSC.2020.2964552

[42] Shili, M., Sohaib, O.: Geographic recommender systems in e-commerce based on population. PeerJ Comput. Sci. **11**, 2525 (2025) https://doi.org/10.7717/PEERJ-CS.2525

[43] Chen, P., Leung, J., Zhang, Y., Shen, Z.: The POWER of ikigai: Optimizing life fulfillment with an integrated user simulator and adaptive hobby recommender. In: 39th AAAI Conference on Artificial Intelligence, pp. 28915–28922. AAAI Press, Philadelphia, PA, USA, February 25 - March 4, 2025 (2025). https://doi.org/10.1609/AAAI.V39I28.35159

[44] West, B.T., Welch, K.B., Galecki, A.T.: Linear Mixed Models: a Practical Guide Using Statistical Software. CRC Press, Florida, USA (2014). https://doi.org/10.1201/b17198

[45] Meyers, L.S.: Applied Multivariate Research : Design and Interpretation / Lawrence S. Meyers,. SAGE Publications, Thousand Oaks (2006). http://www.loc.gov/catdir/toc/ecip0510/2005009519.html

[46] Cohen, L.: Statistical Power Analysis for the Behavioral Sciences, 2nd. edition edn. Lawrence Earlbaum Associates, New Jersey, USA (1988)

[47] Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to Linear Regression Analysis. John Wiley & Sons, Hoboken, New Jersey (2021)

[48] Faul, F., Erdfelder, E., Lang, A.-G., Buchner, A.: G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behavior Research Methods **39**(2), 175–191 (2007) https://doi.org/10.3758/BF03193146

[49] Reguera-Bakhache, D., Garitano, I., Uribeetxeberria, R., Cernuda, C.: An industrial HMI temporal adaptation based on operator-machine interaction sequence similarity. In: 22nd International Conference on Industrial Technology, Valencia, Spain, pp. 1021–1026 (2021). https://doi.org/10.1109/ICIT46573.2021.9453580

[50] Carrera-Rivera, A., Larrinaga, F., Lasa, G., Martínez-Arellano, G., Unamuno, G.: Adaptui: A framework for the development of adaptive user interfaces in smart product-service systems. User Model. User Adapt. Interact. **34**(5), 1929–1980 (2024) https://doi.org/10.1007/S11257-024-09414-0

[51] Chen, X., Tang, X., Zhao, Y., Huang, T., Qian, R., Zhang, J., Chen, W., Wang, X.: Evaluating visual consistency of icon usage in across-devices. International Journal of Human-Computer Interaction **40**(9), 2415–2431 (2024) https://doi.org/10.1080/10447318.2022.2162275

[52] Bedi, P., Goyal, S.B., Rajawat, A.S., Shaw, R.N., Ghosh, A.: A framework for personalizing atypical web search sessions with concept-based user profiles using selective machine learning techniques. Lecture Notes in Networks and Systems **218**, 279–291 (2022) https://doi.org/10.1007/978-981-16-2164-2_23

[53] Wholin, C., Runeson, P., Höst, M., Ohlsson, C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Springer, Heidelberg (2024). https://doi.org/10.1007/978-3-662-69306-3