



# AdaptUI: A Framework for the development of Adaptive User Interfaces in Smart Product-Service Systems

Angela Carrera-Rivera<sup>1</sup> · Felix Larrinaga<sup>1</sup> · Ganix Lasa<sup>1</sup> ·  
Giovanna Martinez-Arellano<sup>2</sup> · Gorka Unamuno<sup>3</sup>

Received: 10 October 2023 / Accepted in revised form: 16 July 2024  
© The Author(s) 2024

## Abstract

Smart Product–Service Systems (S-PSS) represent an innovative business model that integrates intelligent products with advanced digital capabilities and corresponding e-services. The user experience (UX) within an S-PSS is heavily influenced by the customization of services and customer empowerment. However, conventional UX analysis primarily focuses on the design stage and may not adequately respond to the evolving user needs during the usage stage and how to exploit the data surrounding the use of S-PSS. To overcome these limitations, this article introduces a practical framework for developing Adaptive User Interfaces within S-PSS. This framework integrates ontologies and Context-aware recommendation systems, with user interactions serving as the primary data source, facilitating the development of adaptive user interfaces. One of the main contributions of this work lies on the integration of various components to achieve the creation of Adaptive User Interfaces for digital services. A case study of a smart device app is presented, to demonstrate the practical implementation of the framework, with a hands-on development approach, considering technological aspects and utilizing appropriate tools. The results of the evaluation of the recommendation engine show that using a context-aware approach improves the precision of recommendations. Furthermore, pragmatic aspects of UX, such as use-

---

✉ Angela Carrera-Rivera  
aicarrera@mondragon.edu

Felix Larrinaga  
flarrinaga@mondragon.edu

Ganix Lasa  
glasa@mondragon.edu

Giovanna Martinez-Arellano  
giovanna.martinezarellano@nottingham.ac.uk

Gorka Unamuno  
gunamuno@ideko.es

<sup>1</sup> Faculty of Engineering, Mondragon Unibertsitatea, Loramendi 4, 20500 Arrasate, Spain

<sup>2</sup> Institute for Advanced Manufacturing, University of Nottingham, Nottingham, UK

<sup>3</sup> ICT and Automation Research Group, IDEKO, Elgoibar, Spain

fulness and system efficiency, are evaluated with participants with an overall positive impact on the use of the smart device.

**Keywords** Smart Product–Service Systems · Context-awareness · Adaptive User Interface · Recommendation System

## 1 Introduction

Smart Product–Service Systems (S-PSS) are digital ecosystems that integrate stakeholders, smart devices, and platforms, and are characterized by their high complexity (Kuhlenkötter et al. 2017; Liu et al. 2018; Pirola et al. 2020). As a business strategy, S-PSS combine smart products with digital services that should be capable of capturing the needs of multiple stakeholders by following a “value co-creation” view, a fundamental pillar of Service Design that uses experiences, context, and multi-stakeholder participation to create innovative business value propositions (Wetter-Edman et al. 2014). Providers can generate richer experiences for customers by understanding their preferences, and a deep understanding of customer experiences could help companies define and redefine better value propositions (Pezzotta et al. 2017). User experience (UX) can be a clear differentiator among competitors to achieve economically sustainable PSS (Seo et al. 2016). However, the design of S-PSS and UX analysis in multiple studies is only limited to the early phases of the product development and they do not address the evolution and adaptability that S-PSS should have. To create effective S-PSS, user behavior and preferences should be associated with different design elements in specific usage contexts (Cong et al. 2020). In a related study, Yu and Sung (2023) conducted an investigation into the adoption intention of S-PSS, with a specific emphasis on smart appliances. Their research involved a survey with a sample of over 500 participants, and the results revealed a robust correlation between the adoption and usage of S-PSS and the perceived usefulness and flexibility of these systems in catering to users’ specific needs.

The UX is linked to the users perceptions and responses during their interactions with the product and service, as stated by Dong et al. (2019). The effectiveness of S-PSS UX is determined by how well the interaction journey aligns with and fulfills user needs and expectations. The use of digital platforms within the S-PSS serves as communication hubs between users and smart products, and related services (Fig. 1). For instance, Bosch, a renowned technology company, employs a mobile app to deliver a diverse range of e-services designed specifically to enhance the functionality of their smart home devices (Bosch 2022). Lenze, a leader in automation technology, integrates an app to manage and optimize services associated with their smart motors (Lenze 2022). In the market of consumer and e-health technology, Fitbit enhances its offerings by providing a smart training band alongside a mobile app. This app not only supports fitness tracking but also encompasses a diverse range of services (Fitbit 2023). These examples demonstrate how these digital platforms become integral components of the overall value proposition. Services provided can be directly linked to the device, but they can also exist independently showing the diversity on the aspects

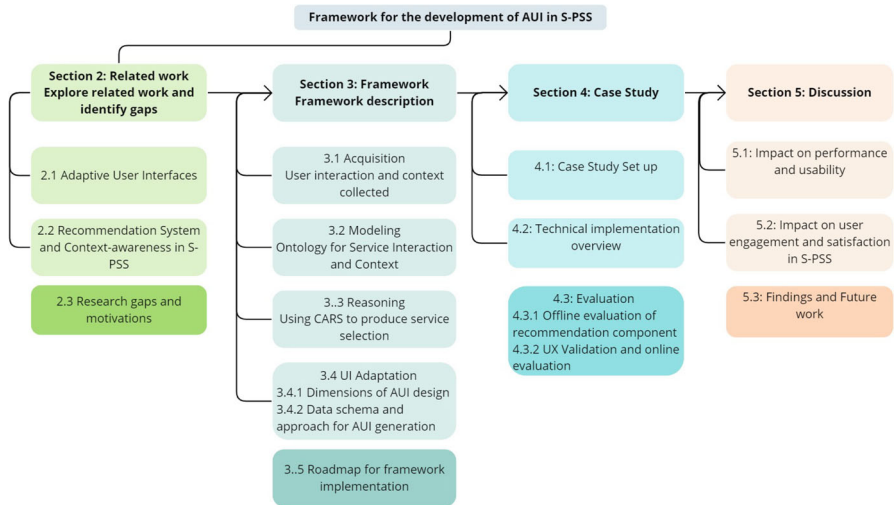


**Fig. 1** Digital platforms on the S-PSS ecosystem **a** Bosch, **b** Lenze, **c** Fitbit

of user experience within the S-PSS ecosystem. Therefore, User Interfaces within these platforms play a fundamental role in shaping the UX.

However, Carrera-Rivera et al. (2022) identified that current research does not address how to exploit the data generated from devices and users to provide customized experiences and adaptable design capable of reacting to different sources of context (i.e., device context, user context, and environment context) as S-PSS have a highly context-dependent nature (Valencia Cardona et al. 2014). Similarly, Cong et al. (2020) emphasized that “user preferences should be associated with different design elements of S-PSS in specific usage contexts.” While existing research primarily focused on customer data from surveys or data gathered from online reviews on social media, there is a need for further work on adapting the means of interaction with S-PSS to enhance the user experience and cater to individual user needs by using data surrounding the usage of digital services (Liu et al. 2018).

In the literature, multiple authors have studied user interactions to understand user tasks and goals in fields such as engineering design (McComb et al. 2017) and others (Champiri et al. 2019). This article proposes the exploitation of user interactions in conjunction with contextual data to adapt the way in which services are presented to the user. It introduces a framework that generates an Adaptive User Interface (AUI) based on user interactions and context as valuable indicators of user behavior while utilizing a Recommendation System as an engine to generate personalized suggestions



**Fig. 2** Graphical description of the article

on services and User Interface Descriptor (UID) and a server-driven approach for real-time User Interface (UI) adaptations

The paper organization is depicted in Fig. 2. Section 2 reviews background and related works to define research gaps and motivations. Section 3 firstly presents the framework that guides this development approach with a detailed explanation of each of its component. Moreover, a roadmap for framework implementation is presented. Then, in Sect. 4 a case study is set up using a smart coffee vending machine as a case study of an S-PSS that serves as a practical demonstration and evaluation of the framework. Section 5 presents the discussion of the findings and limitations drawn from the study and opportunities for future work. Finally, Sect. 6 presents the conclusions of the study.

## 2 Literature review

S-PSS represent a novel business approach that has emerged in recent years, where a Smart Connected Product (SCP) offers a range of ICT-enabled services, allowing users to utilize the product's functionality and exchange information during its *usage* phase (Chou 2021). It can be defined as a digital-enabled business solution supplied within an ecosystem, which provides economic and sustainable value to the customer. This value is achieved by integrating intelligent products with data-enabled services made possible by physical and digital infrastructures (Pirola et al. 2020). Characterized by a high level of digitalization, S-PSS leverage intelligence capabilities, ubiquitous connectivity, and analytic capabilities to deliver personalized and convenient experiences, continuously co-creating value with stakeholders (Wang et al. 2019).

To design an S-PSS, scholars emphasize the importance of following a closed-loop design approach. This method enables continuous iteration and self-adaptation

throughout the entire lifecycle in a holistic manner (Wang et al. 2019; Cong et al. 2020). The closed-loop design approach facilitates active participation from both customers and providers, who contribute directly or indirectly in different phases of the design process. For instance, during the requirements analysis phase, customers can act as co-ideators, by identifying and collecting user requirements and contributing to the design of the product. During the innovative design phase, customers can provide feedback on prototypes. Seo et al. (2016) proposed a hybrid evaluation method for smart home services based on VR and AR prototypes. During the design evaluation phase, they can act as co-evaluators by evaluating the value proposition and customer value of the product. Wang et al. (2020) proposed an approach for the concept evaluation of service bundles using sentiment analysis of customer opinions. Finally, during the usage phase, users can take on the role of co-innovators by suggesting improvements or adjustments to the product. Alternatively, users can have the role of experience creators, as producers can generate richer experiences for a set of customers and, in return, access their latent perceptions and preferences, capitalizing on them. A deep understanding of the user experiences not only helps firms in defining better value propositions but also supports improved customization of products and services. (Pezzotta et al. 2017).

UIs within S-PSS serve as cognition and communication platforms linking humans and devices, facilitating information exchange among users, products, and services (Gong 2009). Responsive interfaces, adapting to user behavior, preferences, and contextual changes, effectively optimize UX. In S-PSS, adaptive interfaces could dynamically tailor user experiences in real time, aligning interactions with individual needs, thus enhancing satisfaction and engagement, crucial for successful S-PSS implementations.

## 2.1 Adaptive User Interface (AUI)

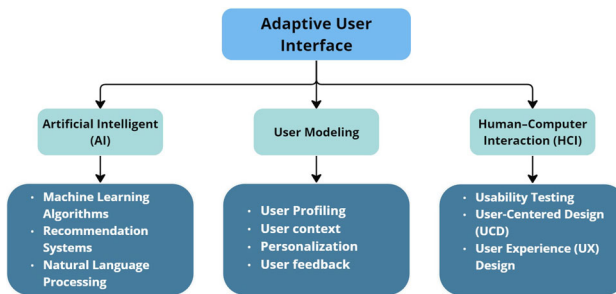
An AUI serves as a dynamic component, capable of altering interface characteristics and functionality based on user requirements. Also referred to as “Intelligent User Interface” in more recent studies (Brdnik et al. 2022), AUIs have played a central role in the realm of Human–Computer Interaction (HCI) since its inception (Mitchell and Shneiderman 1989). Miraz et al. (2021) undertook an extensive literature review on the topic of AUI. They found that one of the principal issues is the lack of a clear definition of the constituents of User Interface adaptation. Various definitions of the term found in the literature are presented in Table 1. Based on this, we conclude that an AUI is *an intelligent UI that utilizes Artificial Intelligence techniques, such as machine learning, Recommendation Systems, natural language processing, or others, to dynamically adjust one or more UI elements in real time. It responds to the current context of use with the goal of enhancing usability.*

The three pillars of AUI development are Artificial Intelligence (AI), User Modeling, and Human–Computer Interaction as presented in Fig. 3.

The landscape of AI encompasses a diverse array of tools, techniques, and algorithms. In the development of AUIs, various AI approaches come into play. One of the earliest examples is presented by Stephanidis et al. (1998), this work introduced

**Table 1** Definitions Adaptive User Interface (AUI)

Definitions	References
“...and intelligent and adaptive means to implement the proper response techniques. Furthermore, the intelligent interface can learn about the subject’s lifestyle and adapt the interaction style/mode accordingly”	(Orghidan et al. 2013)
“Human–machine interfaces that have the objective of improving the effectiveness, naturalness and the efficiency of interaction (human–machine interactions) by reasoning (involving Artificial Intelligence methods), representing and modeling the users, tasks, device and context”	(Tahir 2015)
“Human–machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human–machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse. They use Artificial Intelligence (AI), human–computer interaction (HCI), software engineering (SE) and other techniques to promote more natural and usable human–machine interaction”	(Gonçalves and da Rocha 2019)
“...by their capability to adapt at run-time and make several communication decisions concerning ‘what’, ‘when’, ‘why’ and ‘how’ to communicate”	(Völkel et al. 2020)
“An intelligent User Interface is a UI that contains some perspective of Artificial Intelligence in computing. This makes the interface more comprehensive, customizes and guides the interaction”	(Wenjuan et al. 2022)

**Fig. 3** Adaptive User Interface pillars

a conceptual methodology aimed at dynamically adjusting the UI to individual user needs, particularly in terms of accessibility, using a rule-based approach.

Machine learning techniques and rule-based systems stand out as two of the most common approaches in AUI development, relying on predefined rules to govern how the interface should adapt under specific conditions. For instance, Hussain et al. (2018) introduced a rule-based AUI tailored for a wellness application. Their methodology/system is centered on contextual data and an authoring tool, allowing for personalized interface adjustments that provide users with an enhanced and tailored experience.

Similarly, Khan and Khusro (2020) presented a model-based and context-aware approach for smart cars. This AUI aimed to mitigate driving distractions by considering contextual information in the model and user interactions to create rules. In a different vein, Stefanidi et al. (2022) developed a context-aware AUI for augmented reality interfaces, specifically designed to assist law enforcement agents. Their system leveraged user contextual data and semantic reasoning through SWRL rules and delivered dynamic adaptations that improved usability.

Alternatively, some AUIs adopt a recommendation-based strategy (Oestreich et al. 2022). These systems rely on Recommendation Systems (RS) and Context-aware recommendation system (CARS) that often integrate machine learning algorithms. They analyze user behavior and context to offer recommendations for interface adaptations, which users are able to accept or decline.

Another challenge to adaptation is that there is no unified software architecture or framework to support lifecycle development of AUI. In a systematic mapping review on the topic of Intelligent User Interfaces and AUI from 2012 to 2022 (Brdnik et al. 2022), the research identified 212 studies, among these only ten corresponded to frameworks and four to methodologies (Table 2). An observation also found in recent research (Ali et al. 2024) reinforces the existing gap in the field; many studies in this domain have failed to conduct a validation process for their proposed models or frameworks and have often focused exclusively on a single specific product. Machado et al. (2018) introduced a conceptual framework tailored for real-time adaptive interfaces in web-based applications, with a specific focus on addressing challenges commonly encountered by older individuals, such as vision loss and cognitive decline. Similarly, Watearachchi et al. (2020) proposed an adaptive framework that recommends various functions based on users emotions and contextual parameters, including location, time, and current user behavior, aiming to enhance the overall UX. Notably, their framework emphasized facial expressions. In contrast, Yigitbas et al. (2017) presented a model-driven Integrated Development Environment (IDE). This IDE offers integrated views for UI, context, and adaptation modeling based on rules. Additionally, the authors substantiated the effectiveness of their IDE through the design and validation of a case study.

Table 2 shows methodologies or frameworks found on the literature. Ten of the fifteen described studies do not include any kind of UX or usability evaluation. These studies correspond to “solution proposals” or “validation research.” Five studies include either UX or usability evaluation, but not both. Usability evaluation focuses on task-related aspects such as effectiveness and efficiency, while UX considers overall satisfaction and both pragmatic and hedonic goals. It is also important to highlight that some studies focus solely on Human–Computer Interaction aspects without integrating technological aspects, and vice versa.

## 2.2 Recommendation System and context awareness in S-PSS

As described in the previous section on the development of AUIs, there is a need for an AI technique capable of providing adaptation. In this context, the framework employs the Context-aware recommendation system (CARS). S-PSS are significantly impacted

**Table 2** Frameworks (F) and methodologies (M) published in the Adaptive User Interface (AUI) literature with AI technique & modeling

Study	AI technique & Modeling	Solution offered	Research Type	Domain	Intelligent entity	UX	Usability	Method
Stefamidi et al. (2022)	SWRL Rules and Darlene ontology model	M	Evaluation research	Software	Interface	Yes	Yes	Expert-based evaluation, User testing
Gil et al. (2021)	-	F	Evaluation research	Software engineering	Interface	No	Yes	User testing, Questionnaire
Abraão et al. (2021)	Not specified	M	Solution proposal	Human-computer	Interface	N/A	N/A	
Berman et al. (2021)	Not specified	M	Validation research	3D printing	Interface	No	Yes	User testing, Questionnaire
Yigitbas et al. (2020)	Rule-based and Context and Adaptation model	F	Validation research	Software engineering	Interface	No	Yes	Integrated survey
Gan et al. (2020)	Bayesian framework	F	Validation research	Human-computer	Model for gesture prediction	N/A	N/A	Interaction research
Wallace (2020)	-	F	Validation research	Software engineering	Interface	Yes	No	Automated testing research
Stumpf (2019)	Theoretical Framework	F	Solution proposal	Human-computer	System	No	No	Interaction research
Tan et al. (2018)	C5.0 Decision tree and model UI templates	F	Validation research	Software engineering	Layout Generation for multiple devices	No	No	Interaction research
Iqbal et al. (2018)	AI not specified. Design criteria	F	Solution proposal	Human-computer	N/A	No	No	Interaction research
Landowska et al. (2016)	Design methodology	M	Validation research	Software engineering	Video Games	N/A	N/A	
Zhou et al. (2015)	Classification algorithms	F	Validation research	Human-computer	N/A	N/A	N/A	



Table 2 continued

Study	AI technique & Modeling	Solution offered	Research Type	Domain	Intelligent entity	UX	Usability	Method
Peck et al. (2015)	Weka library of machine learning algorithms	F	Solution proposal	Software engineering	System	No	No	
Fernandez-Garcia et al. (2015)	Not specified. User model	M	Solution proposal	Software engineering	Interface	No	No	Illustrative Example
Mezhoudi et al. (2015)	Design approach	M	Validation research	Human-computer	Interface	No	Yes	Interaction research

by the context in which the services and products are used. Several researchers have highlighted the importance of “contextual data” in S-PSS (Cong et al. 2020; Zheng et al. 2019) because it helps ensure that services and products can adapt in multiple scenarios and to various stakeholders.

Recommendation Systems (RS) are intelligent algorithms that analyze user preferences and behavior to provide personalized suggestions or recommendations for products, services, or content (Jannach et al. 2010). In general, the goals and uses of RS in the S-PSS context would be:

- Enhancing the user experience: Provide personalized recommendations for services based on the user’s preferences, history, and context.
- Increasing the adoption of services: Promoting less-known services to users that are relevant in a given context, increasing their adoption and usage.
- Improving customer retention: Provide personalized recommendations to users that keep them engaged with the PSS application.
- Maximizing the value of the PSS: Suggesting the most relevant and valuable service configuration to the user, increasing the overall value of the PSS for the user (Zhang et al. 2023; Chiu et al. 2021).
- Providing instructional support: Suggesting relevant resources to help users to complete tasks more effectively.

CARS are a type of recommendation engine that consider contextual information to provide personalized recommendations to users. In general, a context-aware system uses context to deliver information and/or services relevant to a user’s task (Abowd et al. 1999). Context refers to any data that can be used to understand the current state, whether situational or locational, of an entity (Ochoa et al. 2023). Context can be categorized from both operational and conceptual perspectives. The operational perspective categorizes context based on how it is acquired, distinguishing between primary and secondary sources where primary will refer to any information retrieved without any kind of pre-processing or additional operations. The conceptual perspective categorizes context based on meaning and relationships. Abowd et al. (1999) introduced one of the first categorizations, identifying four main groups: location, identity, time, and activity. Table 3 presents examples from both operational and conceptual perspectives, illustrating a level of generalization when contextual information is obtained through a secondary process.

A great part of the efforts in the context-aware applications for S-PSS have been observed in the concept evaluation of product–service bundles, and requirement elicitation and analysis for capturing implicit requirements, for which it has been explored external sources to smart products such as social sensors. Several case studies coincide in the analysis of user comments related to their experiences, obtained through social networks using NLP techniques, and the evaluation using sentiment analysis to quantify the positive or negative feelings from text (Mourtzis et al. 2018; Wang et al. 2021; Le et al. 2020; Li et al. 2020; Lin et al. 2016). But context can be also used to resolve what services or information needs to be presented to the user. Therefore, context awareness capability can be used to personalize and adapt the UX by providing services for a specific user situation. This adaptation is driven by a system’s ability to

accommodate individual users’ physical and mental abilities, as well as the context of use and platform capabilities (Stephanidis 2001).

Table 4 presents a summary of context-aware case studies relevant to S-PSS from the literature, offering insights into their objectives, data sources, applied techniques, and the specific stages in the S-PSS lifecycle where the case study or approach was applied.

According to Adomavicius and Tuzhilin (2010), CARS have become more popular due to the growing availability of contextual information and the need for personalized and relevant recommendations. By leveraging contextual information, these systems can provide more accurate and useful recommendations to users, leading to increased user satisfaction and engagement. In CARS, there are three fundamental approaches for integrating contextual data into recommendation algorithms, each dependent on the phase at which contextual data are processed (Papadakis et al. 2022). The first approach is *contextual pre-filtering*, where contextual data are utilized to filter the data prior to the recommendation model application. Alternatively, in *contextual post-filtering*, traditional recommendation algorithms compute preferences based on the entire dataset, and then, the resulting recommendations are filtered according to the user’s context. Lastly, the *contextual modeling* approach involves the direct integration of contextual information into the recommendation model, such as incorporating it into the preference computation process.

In the literature, RS have been used in S-PSS for different purposes. Esheiba et al. (2021) proposed a hybrid knowledge-based RS that recommend the customer a set of customization requirements and design parameters concerning a product for a new PSS configuration from previous service blueprints. Similarly, Chiu et al. (2021) developed a personalized Recommendation System for S-PSS based on an unsupervised learning model, using customer journey maps and user reviews on external websites, to generate recommendations. These approaches are valuable in the *design* stage to understand user requirements and create new service configurations for a smart product. However, relying solely on these sources becomes difficult once the smart product is in use, as it requires active user participation or scraping external sites.

**Table 3** Context categorization based on conceptual and operational perspectives

Conceptual/operational Identity	Location	Time	Activity	Smart Connected Product	
Primary	User ID (based on NFC tag)	GPS coordinates	Timestamp from device	Data from user interactions	Battery voltage
Secondary	User role	Place name	Employee shift schedule	Most visited apps based on interactions	Battery level (i.e., full, medium, low)

**Table 4** Context-aware case studies found in the literature relevant to S-PSS

Case study	Objective	Data used	Technique	Lifecycle stage
(Wang et al. 2021, 2022) 3D Printer	Hypergraph-based smart PSS configuration that enhances customization by incorporating usage scenarios	3Dhub.com user comments and Experts. Product-service bundles of 3D printer	Supervised machine learning (SVM), NLP	Smart design-usage. Concept validation (users as co-evaluators)
(Mourtzis et al. 2018) Mold-die-making industry	Lean PSS evaluation framework integrates KPIs, sentiment analysis, and Lean rules for sustainable design improvement and customer satisfaction	Comments on issues and maintenance and manufacturing data from sensors	Supervised machine learning (SVM), NLP	Smart design-usage. Concept validation (users as co-evaluators)
(Wang et al. 2019, 2021) Framework evaluation using open data to simulate a use-oriented bicycle PSS	Data-driven graph-based requirement elicitation for Smart PSS using ICT and user-generated data for design improvements and value creation	Reviews, comments on third-party apps and sensor data	Graph algorithms	Smart design-usage. Requirement elicitation (users as co-ideators)
(Chiu et al. 2021) Rideshare PSS	Introduces personalized Smart PSS method using NLP for customer requirements, validated via tourist recommendations	Google Maps API, user location	Item-based collaborative filtering	Smart usage

Table 4 continued

Case study	Objective	Data used	Technique	Lifecycle stage
(Seo et al. 2016) Smart home	Hybrid reality-based user experience for context-aware smart homes using VR, AR, and world-in-miniature (WIM) prototyping	VR devices and AR app interactions	Semantic reasoning (Jean inference)	Smart design, Smart prototyping (users as co-testers)
(Esheiba et al. 2021) Laser machine for turbine engine manufacturer	Hybrid knowledge-based recommender system for personalized selection of PSS variants in mass customization	Previously customized PSS variants maintained in a blueprint knowledge base	Ontology-based recommendation and Weighted similarity (Euclidean distance)	Smart design (Configuration of new PSS)
(Cong et al. 2022) Surgical robot	Machine learning-based iterative design approach to automate user satisfaction prediction in the Smart PSS environment	Task data from 480 samples. User satisfaction surveys from 20 participants	SVM, decision trees, and BP neural networks	Smart Design (users as co-evaluators)
(Cong et al. 2022) Smart 3D printer platform	Context-aware diversity-oriented knowledge recommendation for smarter engineering solution design	Sensor data from 3D printer. User historical behavior records (2638 records)	Item-based Collaborative Filtering using KNN and Pearson correlation	Smart design and Usage
(Yuan et al. 2023) Sporting event service	Proposes a conceptual framework for an S-PSS with context awareness to produce the appropriate service contents for each user modality level	Physical state of users based on wearable devices	Deep neural network model	Smart design and Usage (product-service bundle configuration)

## 2.3 Research gaps and motivations

From the preceding review, several observations have emerged that reveal research gaps in the literature.

- Limited research has focused on the 'usage' stage in the S-PSS lifecycle, where User Interfaces facilitate communication between users and smart products. Real-time User Interface Adaptation, which leverages internal data sources to respond to individual user needs, remains an underexplored area with the potential to significantly enhance user experiences.
- Greater emphasis needs to be placed on the exploitation of internal data sources within the S-PSS, specifically focusing on fulfilling the role of the 'user as experience creator' in a data-driven manner. This entails considering data sources that enable the identification of user preferences and perceptions.
- The absence of software architectures or frameworks for AUI development is a significant challenge, as revealed in a recent studies emphasizing the need for frameworks and methodologies (Brdnik et al. 2022). Furthermore, current research often lacks comprehensive validation processes for proposed AUI models, tending to focus more on single-product solutions (Ali et al. 2024).

To address the above gaps, this paper proposes a framework for real-time User Interface Adaptation and the development of applications that dynamically adapt User Interfaces to user behaviors and contextual changes within the S-PSS ecosystem. This framework aims to contribute to the enhancement of user experiences of Smart Product–Service Systems.

## 3 Framework

This work presents a framework designed to serve as a conceptual structure for organizing and guiding system development. A framework offers predefined components and guidelines, providing a structured approach. In this context, the framework builds upon the four general components of the context life cycle, as defined by Perera et al. (2013), and extends its focus to address user interface (UI) adaptation within S-PSS.

Figure 4 depicts the framework, highlighting different stages in the context life cycle and their relation to User Interface Adaptation. The first step is *Acquisition*, which refers to data that need to be acquired from different sources. Specifically, we leverage user interactions as a valuable internal data source, which can be more readily accessible during the usage stage compared to other data sources. Interactions can reveal user behavior when engaging with digital services offered by smart products. In the *Modeling* stage, the collected data are structured and represented in a meaningful way, facilitating its interpretation and analysis. The next stage, *Reasoning*, involves processing the data and context to derive information that can be used to enhance the user experience. Finally, in the *UI Adaptation* stage, the results obtained from the recommendation engine are utilized to generate the User Interface for the end-users of the S-PSS (Carrera-Rivera et al. 2022). By leveraging context awareness, the proposed

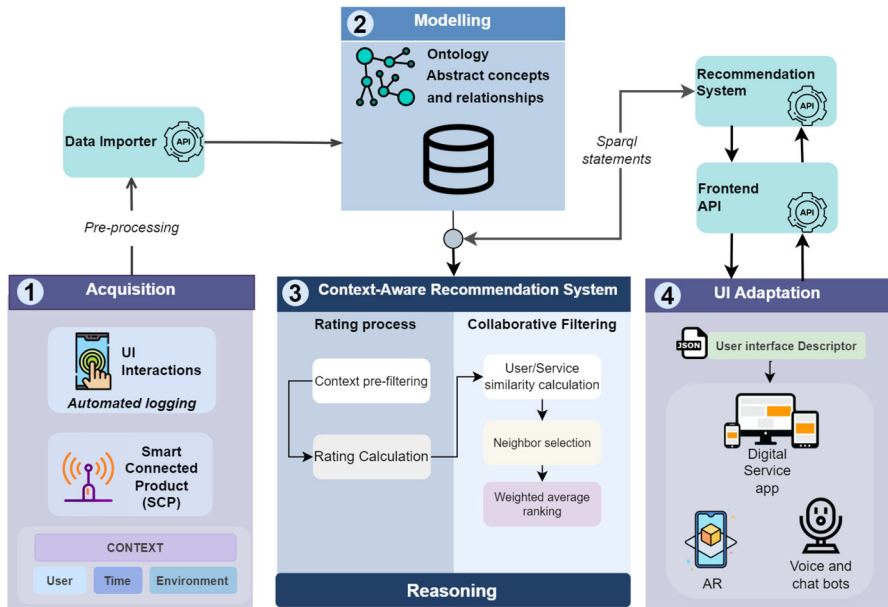


Fig. 4 Context-aware framework for adaptive UX

framework aims to enhance the user experience of S-PSS by providing a personalized way to present services to users based on their current context.

### 3.1 Acquisition

Acquisition refers to the process of collecting data from Smart Connected Products (SCPs) and data relevant to the user and how users interact with products (Pulikottil et al. 2023). SCPs are equipped with sensors, embedded systems, and communication capabilities that enable them to gather various types of data, such as sensor readings, usage patterns, and environmental information. On the other hand, as a way to understand users, interactions can help to infer user behavior. The process of acquiring user interactions can vary depending on the way users interact with the machine or product, usually associated with the visual elements of the interfaces. The framework highlights the use of automated logging of user interactions as the preferred data collection method, offering several advantages such as non-intrusiveness, real-time data capturing, and minimizing errors associated with manual data entry. Specialized software like Matomo or Google Analytics can be used for web or mobile apps that interact with the S-PSS, while custom logging tools can be built directly onto the system. However, this approach can be challenging for complex interfaces and may require access to the system’s underlying code. For instance, Zhou et al. (2019) presented an approach for monitoring back-office staff using a screen-mouse-key logger that captured images, mouse, and key actions, along with timestamps, and was transformed into a UI log through image analysis techniques.

From the acquired data, contextual information relevant to the use of digital services offered by the product can be obtained. The SCP itself is also a valuable source of contextual information that can help provide adaptation, such as alarms, status, and measures of sensors and actuators, potentially affecting how users interact with the machine. However, for adaptation purposes, it is often beneficial to provide a level of generalization in the contextual data to reduce data sparsity that can arise from having overly specific contextual information.

### 3.2 Modeling

The modeling component allows the representation of the data acquired from the S-PSS in terms of attributes, characteristics, and relationships with previously specified context (Liu et al. 2011). To achieve this, ontologies serve as a means to formally model concepts from a specific domain by providing a detailed specification of entities, their properties, and their relations (Guarino et al. 2009). They effectively define the vocabulary associated with the respective domain, enabling a structured representation and organization of knowledge. To start with the construction of the ontology, the UPON methodology (De Nicola et al. 2005) serves as a guide for ontology building derived from the Unified Software Development Process. In the beginning, it was important to define the goal of the ontology, which is “to represent digital services offered by an S-PSS and support the adaptation of User Interfaces” and then define some competency questions (CQs). A CQ describes the type of question an ontology is expected to answer. Therefore, the ontology’s ability to answer these CQs is a crucial functional requirement and a way to evaluate the ontology (Ren et al. 2014). Three CQs are defined and presented below:

CQ1: In which contexts is a service executed?

CQ2: Who are the users of a service or its sub-services?

CQ3: Which interactions execute service  $X$ ?

A hybrid approach is employed to formally define the concepts and relationships among the entities in the ontology (Fig. 5). In the top-down approach, concepts go from general to particular. The methodology highlights the benefits of reusing existing ontologies. From previous analysis, it was identified those ontologies that model the S-PSS main concepts and contextual information specifically for AUI. In the bottom-up approach to ontology design, the concepts in the ontology are derived through the analysis of interaction data instances using datasets and databases.

The resulting ontology to model Service, Interaction and Context (SICO)<sup>1</sup> was created in Protégé, and a view of the general structure is presented in Fig. 6. To visually represent the components of the ontology, we divided the entities into three groups: S-PSS, context, and interaction.

**S-PSS:** In an S-PSS, a static model is necessary to represent the services of smart devices. In accordance with Maleki et al. (2018), each service is modeled into a pattern that relates the service with the product and required information. Services may be

<sup>1</sup> <https://mu-sse.github.io/adaptiveUIs-project/SICO/index-en.html>



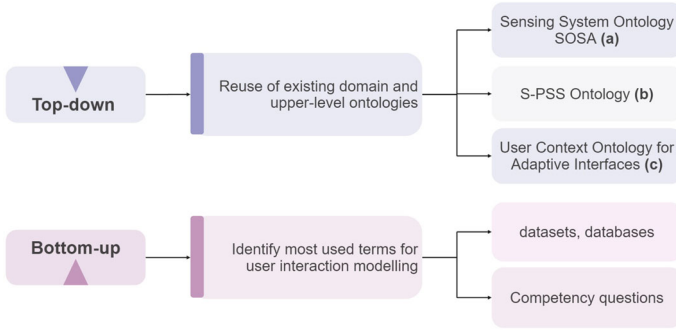


Fig. 5 Ontology design hybrid approach. **a** Janowicz et al. (2019), **b** Maleki et al. (2018), **c** Iqbal et al. (2021)

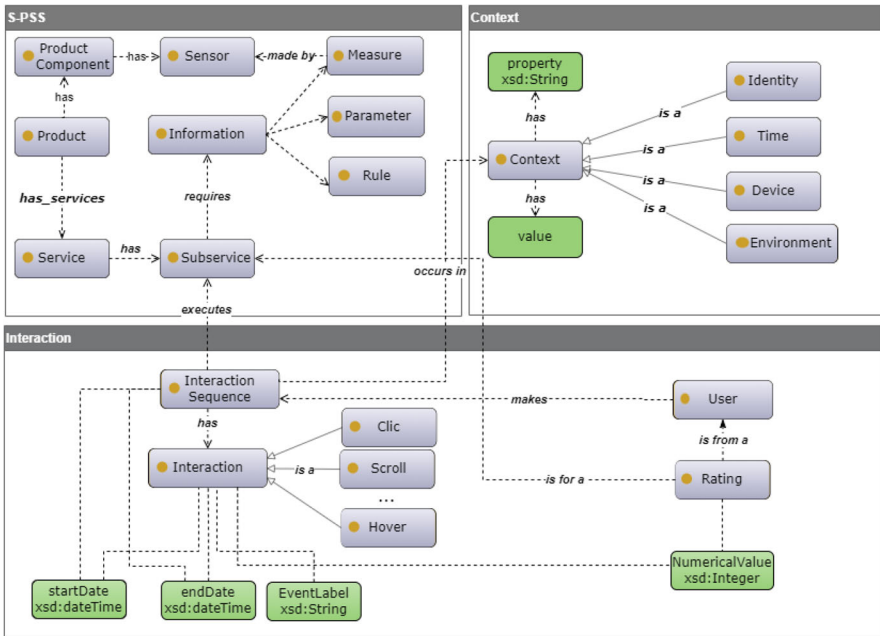


Fig. 6 Ontology general view

grouped into related sub-services. A sub-service represents a specific function of the e-service platform or smart device, requiring information in the form of rules, measures, and parameters. The product itself is closely related to the physical components that need to integrate sensors in the solution.

**Context:** The second group further extends the framework ontology model to incorporate context-specific entities and relationships. The ontology encompasses four major sources of contextual information: device, user/identity, environment, and time (Iqbal et al. 2021). Device context will refer to information relevant to the status and particular properties of the SCP. The user identity context can encompass demographic

details such as age-group, location, education level, reflecting the type of an individual or a user group. The environmental and time context will refer to the surrounding conditions and temporal aspects influencing the SCP or user interaction. This includes factors such as ambient conditions, time of day, and other situational variables that contribute to an understanding of the overall context in which the device or user operates. Because contextual data are dynamic, it needs to be created as instances or individuals derived from these types for each particular implementation of the framework.

**Interaction:** In the last group, following a bottom-up approach, we analyzed a dataset from collected user interactions (Carrera-Rivera et al. 2023), to define the entities and their relationships with the services and contextual data within the ontology. Interaction sequences represent an ordered set of individual interactions performed by users within the system and are linked to a specific service or sub-service. Interactions can take various forms (i.e., click, scroll, hover) and have the ability to trigger events. To enhance the understanding and analysis of these events, it was crucial to name them appropriately. Once these elements were incorporated into the ontology, a comprehensive representation of user interactions and their connections to the system behavior and functionality was achieved.

The proposed framework facilitates the connection between Data Acquisition and Modeling is facilitated within the framework through a *data importer* component. This component is responsible for pre-processing user interactions and transforming them into ontology individuals, which can be further utilized in the system. The pre-processing stage considers those interactions that demonstrate a certain level of engagement with a particular service. This ensures the generation of valid sequences of interactions linked to digital services. A sequence of interactions  $S = \langle e_1, e_2, \dots, e_m \rangle$  ( $e_i \in S$ ) is an ordered list of events  $e_i$  occurring in a user session, where  $S$  is a set of events known and the order is defined by  $i$ . This means that the event  $e_i$  occurs before the event  $e_{i+1}$  and  $E$  must contain at least two events  $e$  to be accepted as a sequence (Reguera-Bakhache et al. 2020). To facilitate further analysis, each type of event is associated with a UI actionable element and it is assigned a numerical value based on the actions triggered. The set of generated sequences is linked to a specific service in the application, enabling a comprehensive understanding of user interactions within the context of that service.

The use of ontologies provides a flexible way for interrelating elements in a natural manner, unconstrained by specific data types. They also enable the derivation of meaningful inferences from established relationships, a feature difficult to achieve in traditional database design.

### 3.3 Reasoning

The reasoning stage involves processing user and contextual data obtained from the use of smart products. These data are then used to create personalized User Interfaces that adapt to each individual. In this context, RS serves as an approach that can leverage this data for this purpose. Collaborative Filtering (CF) is a widely used technique within Recommendation Systems that exploit available data containing user preferences and ratings to identify similarities between users and items. The fundamental idea of user-

based CF algorithms is that users with similar reported preferences are likely to share similar interests. These discovered similarities are then utilized to predict missing ratings for items and generate recommendations for users (Papadakis et al. 2022).

However, in contrast with traditional CF scenarios, where ratings are obtained from users more commonly (e.g., star ratings in e-commerce), for S-PSS this option could be more difficult to achieve, and ratings can be primarily obtained based on user interactions (e.g., clicks, page views, etc.) that reflect the level of engagement with the digital services provided by the product. This approach allows for the generation of implicit ratings, which can be used to uncover similarities between users and/or services. In this work, the implicit ratings, generated through these interactions, are then utilized in the K-Nearest Neighbor algorithm (kNN). kNN is widely employed in Recommendation Systems for CF. It identifies similar users by measuring the similarity of their ratings. By considering a weighted average of the preferences of the  $k$  most similar users, the kNN algorithm predicts a user's preference for a specific service (Kulkarni and Rodd 2020). To implement the kNN algorithm, a matrix containing the computed ratings resulting from both user interactions and explicit feedback serves as the input. The process of generating these ratings will be explained in the following subsection.

### 3.3.1 Context pre-filtering and rating calculation

This work adopts the contextual pre-filtering technique, that is compatible with recommendation algorithms that do not inherently incorporate context awareness (Smirnov et al. 2017). In this approach, context awareness is achieved by transforming the input of the classical recommendation algorithm followed by the utilization of user-based collaborative filtering (Raza and Ding 2019).

Context pre-filtering is a technique that involves using values of contextual attributes as constraints for selecting ratings that will be used in a Recommendation System's user-item space (Kulkarni and Rodd 2020). An important initial aspect to consider in this work is the "context generalization," where contextual attributes are assigned into broader categories (Smirnov et al. 2017). As presented in the ontology, four contextual sources were defined, including *identity*, *device*, *environment*, and *time*. From these sources, contextual values can be grouped together to represent broader contextual categories. For example, specific user demographics (such as age, or job type) can be grouped together to represent a broader category of "user profile." Similarly, specific time periods can be grouped together to represent a broader category, such as "season" or "shift" (see Fig. 8). Generalizing contextual attributes in this way allows for reducing the dimensionality of the contextual space and the data sparsity.

Moving forward, as presented in Fig. 7, the RS API receives a set of generalized contextual data in a JSON format. A SPARQL query is sent to the ontology. In this way, interactions that are irrelevant to the Recommendation System are filtered out beforehand. As defined in the ontology, a single *Interaction* may *occur\_in* multiple *Context* instances (Fig. 8). Therefore, the filter parameters used in the query are non-exclusive to avoid significantly reducing the available data points.

There are two types of feedback associated with the user: implicit feedback which is derived from user actions and behavior and explicit feedback which is obtained

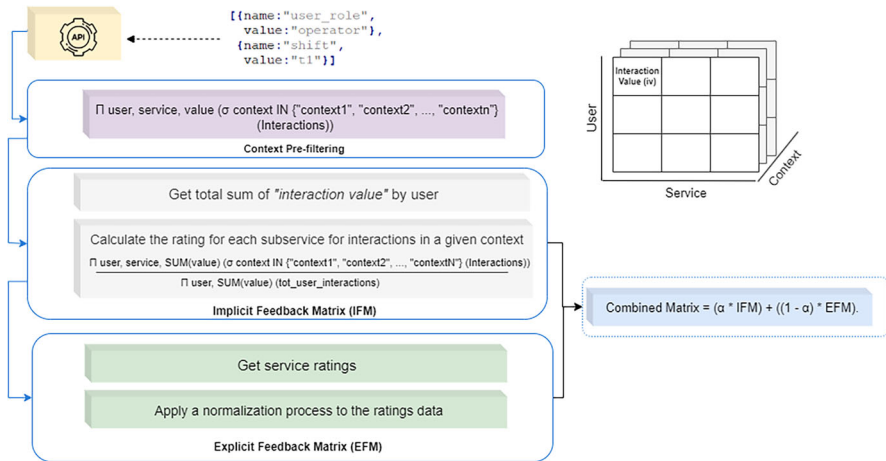


Fig. 7 Rating calculation

through direct input from users and directed to a specific service or item. In this work, each type of interaction is considered a form of implicit feedback and represented as a binary value that is later used to summarize and calculate the ratings. Positive implicit feedback is given a value of **1** that represents a positive engagement with a service (Li et al. 2016). The rating is a measure of user engagement that represents the ratio of the total sum of interaction value a user has made across a given set of contexts to the number of interactions they have made for an individual service. In relational algebra, this can be expressed as:

$$\frac{\prod user, service, SUM(value)(\sigma context IN \{context1, context2, \dots, contextN\} (Interactions))}{\prod user, SUM(value)(tot\_user\_interactions)}$$

This provides a normalized measure of user activity that takes into account the user’s broader range of interactions, allowing for more accurate comparisons between users and services. As stated before, interactions can occur in multiple contexts. Therefore, if the same interaction occurs in more than one context within the set of contexts being filtered, the rating for the service will be calculated as if there were separate interactions. This can result in a higher rating for services that fulfill more than one of the contexts (Nguyen et al. 2016). This approach can help with ‘cold-start’ when there is not relevant information for an specific set of context and reduce the data sparsity problem associated with context pre-filtering. The data become less sparse as each interaction is associated with multiple ratings, which increases the probability that services will be accurately represented in the data. Moreover, this allows to generate more data points, which can help to mitigate sparsity by providing additional information that can be used to better estimate the true value of the interactions.

On the other hand, when users provide explicit feedback, they assign a score to an item using a predefined scale. Thus, the ratings will not be compatible with the implicit feedback rating. To address this disparity, service ratings should be averaged and

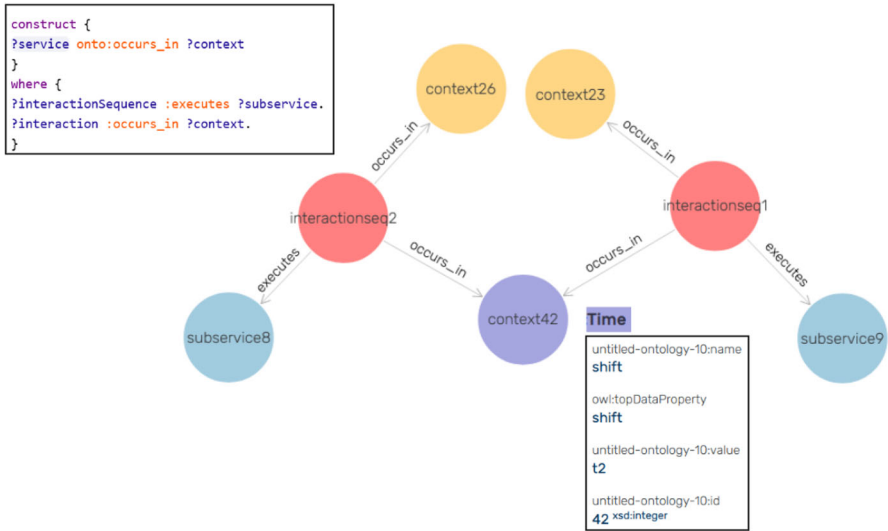


Fig. 8 Querying interaction sequences occurring in several context

normalized to a common scale ranging from 0 to 1 (Liu et al. 2010). This normalization process allows for a unified representation of feedback by combining the implicit feedback matrix (IFM) and explicit feedback matrices (EFM) to generate a combined matrix that incorporates weighted contributions from each type as presented in Fig. 7.

The next step is to compute the similarities among users to predict a recommendation of services. For that purpose, three similarity metrics found in the literature were evaluated: cosine, euclidean, and Pearson similarity, which are commonly used similarity metrics in collaborative filtering systems (Jannach et al. 2010), as they are well suited for high-dimensional data and can effectively capture the similarity between vectors. For instance, Tarus et al. (2017) used adjusted cosine similarity for computing ratings based on ontology domain knowledge and making predictions for the target learner in a hybrid knowledge-based recommender system for the recommendation of e-learning resources to learners. Campos et al. (2010) employed Pearson coefficient and variations in the KNN algorithm, along with an ad hoc strategy, to take into account the temporal context for movie recommendations.

Using these similarity metrics coefficients, we incorporate the kNN algorithm using as input the 2D combined matrix that represents a weighted contribution of both implicit and explicit ratings for each service. Then, limit the neighborhood to predict the weighted average rating of service *s* using the rating given to *s* by users most similar to *u* and returning the *k* more relevant options, ranked by the calculated rating.

$$\text{predicted Rating} = \frac{\sum(\text{sim}(u, s) * r_{u,s})}{\sum \text{sim}(u, s)} \tag{1}$$

### 3.4 UI adaptation

In this stage of the framework to produce the UI adaptation, two factors contribute to the process, an analysis of several dimensions relevant to the design of the AUI and the technical perspective.

#### 3.4.1 Design of Adaptive User Interfaces

The adaptability of User Interfaces requires an analysis of various dimensions that influence user interaction. These dimensions address critical questions, such as *what* elements are adapted in the UI, *when* the adaptation occurs, and the extent of user control over the adaptation process (Oestreich et al. 2022; Abrahão et al. 2021). In the context of S-PSS, a detailed breakdown of these factors is provided in Table 5 and elaborated upon below.

*I. Adaptation Target* Establishing adaptation targets is a fundamental step in digital platforms within S-PSS. The framework focuses on customizing the content based on outputs from the recommendation engine, involving the generation of UI components (e.g., messages, cards, buttons) that represent recommended services or smart device features, especially in mobile and web applications.

Adaptation targets are associated with specific UI elements that will be adjusted in accordance with the recommendations. The subsequent step involves presenting these adaptations to the user, highlighting the significance of adaptation styles.

Adaptation styles play a role in the visual presentation of the selected UI targets (Fig. 9). Adaptation styles described in the literature include (Gobert et al. 2019):

- *Highlighting* This involves the strategic use of a contrasting background color to draw attention to specific elements of the interface. For example, messages or cards representing recommended services can be highlighted, thereby enriching user navigation and comprehension (Tsandilas and Schraefel 2005).
- *Item/Service Re-ordering* The repositioning of selected items provides a streamlined approach to information presentation. This is beneficial for ensuring quick access to frequently used or popular services, enhancing efficiency, and reducing cognitive load (Miraz et al. 2021).
- *Group Reordering* Beyond individual items, entire groups can be reordered within menus, boxes, etc. In this way, related content or functionalities can be clustered, giving rise to a more intuitive and organized User Interface.
- *Folding* Truncating items with lower priority or relevance—commonly referred to as folding—declutters the visual space. This simplifies the User Interface and guarantees that pertinent information is presented to the user, contributing to a more focused and efficient interaction (Bailly et al. 2016).

*II. Initiator of the adaptation* The initiator of the adaptation refers to the mechanism and data employed to start the process of adaptations, which in the case

of the framework is a recommendation-based and context-aware framework. The framework employs user interactions as the main data source, but meaningful contextual sources for the delivery of services will also need to be selected. These can be related to the user, device, task, time, or environment, and are fed into the system.

*III. Moment of the adaptation* The moment of the adaptation dictates when the UI adjustment will occur. In the proposed framework, two moments are considered: (i) when the user starts a session on the device or platform, contextual data are retrieved, and (ii) when contextual data change.

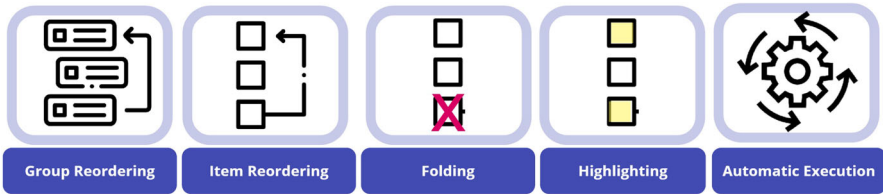
*IV. Control over the adaptation* To analyze the design of the AUI, the amount of control that the user has when interacting with the platform needs to be determined. This is defined by the three sub-dimensions: automation, (decision) participation, and visibility.

In the framework, adaptations are expected to occur autonomously and without user input, as changes in selected contextual data serve as triggers for UI adjustments. This falls within a medium level of automation, as it does not explicitly require user input. However, the framework does not automatically identify when more automation might be necessary, for instance, it does not automatically add new contextual sources.

User participation is derived from explicit and implicit forms, such as accepting or declining recommendations and providing explicit ratings for recommended services as previously explained in subsection 3.3. Therefore, for adaptation to function effectively, the framework requires high user participation.

In terms of visibility, the recommendation-based approach presents a high level of transparency in most cases. However, the perception of visibility of the adaptations can be influenced by the chosen adaptation style. For instance, styles such as “highlighting” or “group reordering” are more visually prominent to users, whereas styles like “re-ordering” or “folding” may be less noticeable. Moreover, the automatic execution of services might go entirely unnoticed by users. Consequently, the latter can feel more natural and may only be noticed if it is not working properly. For instance, if adaptations do not fulfill the needs of the user, and/or no communication is received as to how to modify or revert these changes, the user can suffer negative feelings. This emphasizes the importance of feedback mechanisms and communication in the adaptation process.

*V. Monitoring & Evaluation of the adaptation* The monitoring process of the framework highlights the evolution of the S-PSS. Feedback about decisions made by the application from surveys and user behavior is monitored and evaluated. Three types of evaluation are considered: Offline Evaluation, Online Evaluation, and UX Quality Evaluation. Offline Evaluation involves assessing the performance of the recommendation component based on historical data without directly involving users in real time. Online Evaluation involves the real-time assessment of system performance as users interact with the platform. The UX Quality Evalua-



**Fig. 9** Adaptation styles

**Table 5** Dimensions and descriptions for Adaptive User Interfaces in Smart Product–Service Systems

Dimensions	Description
Adaptation target	Identifies the specific interface elements or aspects for adaptation. For instance, it could involve adapting how a product control panel displays information
Initiator of the adaptation	Determines whether adaptations are triggered by user actions, system algorithms, or a combination of both
Control over the adaptation	Determines the amount of control that should be given to the user <ul style="list-style-type: none"> <li>• Automation: The system’s ability to adapt without user input</li> <li>• Participation: User engagement in adaptation, from minimal to extensive</li> <li>• Visibility: Transparency in how and why adaptations occur</li> </ul>
Moment of the adaptation	Dictates when interface adjustments occur, be it in real-time as users interact, during login, or based on historical usage patterns or changes on contextual information
Evaluation of the adaptation	Involves assessing the effectiveness of adaptive changes through user feedback, performance metrics

Adapted from (Oestreich et al. 2022)

tion of the framework focuses on the more pragmatic and hedonic aspects of the experience.

### 3.4.2 Technical perspective and data schema for Adaptive User Interfaces in S-PSS

From the technical perspective, a User Interface Descriptor (UID) can be used to define the structure and composition of the UI. These descriptors act as data representations to describe UI elements and their layouts and can be written in standardized data exchange formats such as JSON or XML. According to REST application principles, basic UIDs consist of a name, an optional description, and a list of UISections, which are self-consistent modules of the UI, populated by UI element objects (Belli et al. 2015). As a result, each service or featured sub-service in the S-PSS is linked to a UID. They guide the translation of recommendation engine outputs into specific user interface components.



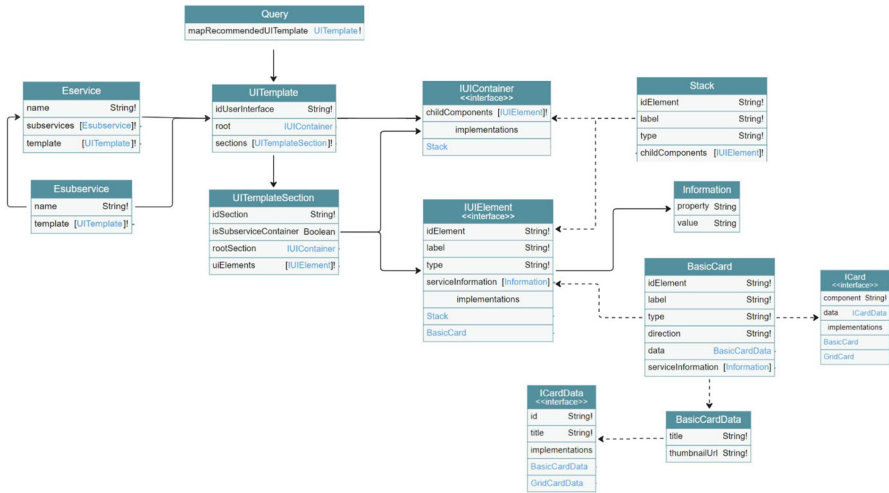
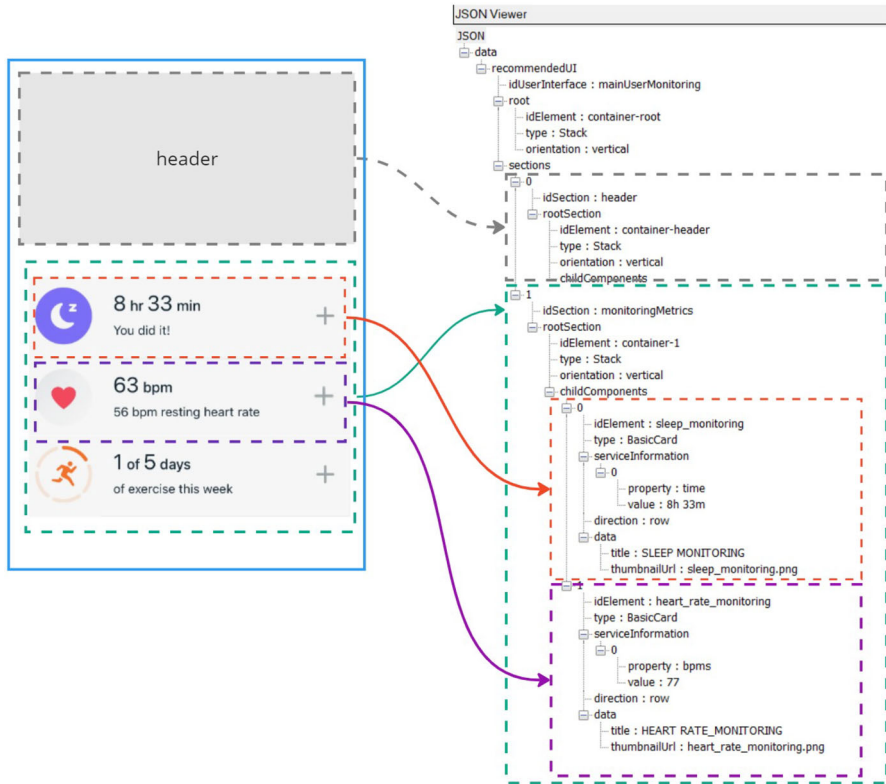


Fig. 10 Schema server-driven UI

The proposed framework employs a server-driven UI design pattern to build the AUI. This approach relies on UIDs sent from the server to the client for rendering. This design pattern offers flexibility by allowing the server to generate the descriptors and dynamically adapting them based on user context and preferences (ThoughtWorks 2022). To support this approach, the framework includes a front-end API, depicted in Fig. 4, which receives recommended services (see Section 3.3.1) and generates the descriptors for the S-PSS user interface based on an schema. This facilitates the process of adapting User Interfaces that are rendered in the S-PSS without needing to redeploy the applications, which allows for a real-time adaptability. This is particularly useful considering that S-PSS are often multi-platform and work on different devices or are even built into the device. The client side includes a rendering mechanism that interprets the UID received from server and generates the appropriate visual elements and layout. Additionally, the client side manages user input and events like clicks or swipes, transmitting them to the server for processing. The schema to generate the UID is presented in Fig. 10 .

The schema introduces a set of interfaces and types that represent the components and data structures of the UI. An interface in the context of data schema and software engineering is an abstract type that defines a common set of fields that any number of object types can employ. Interfaces are used to represent a relationship among different types that have a shared behavior. Then, an object type can use the interface, which is called an implementing object type.

The UIElement interface captures common properties of UI elements such as idElement, label, type, but can also receive serviceInformation. The UIContainer interface represents containers or panels holding child UI components. The UI\_Template type defines a template structure with an idUserInterface, a root container (UIContainer), which represents a panel that can have multiple elements, and an array of template sections (UI\_TemplateSection). The UI\_TemplateSection represents a section within a



**Fig. 11** UI user monitoring for wristband and response from method `mapRecommendedUITemplate` in JSON format

UI template, which can hold multiple `UIElement`s. Specific implementations can be created from these abstract types. For instance, `Stack` is a concrete type of container or panel that is both `UIElement` and `IUIContainer`. `BasicCard` is a concrete implementation of an `UIElement`. Beyond individual UI elements, the schema addresses the connection with E-services, this type represents an electronic service and comprises fields such as name (specifying the service name), sub-services (a list indicating various nested components or sub-services), and template (denoting the User Interface templates associated with this electronic service). Complementing the E-service, the E-sub-service type focuses on individual components or features nested within a larger electronic service. It includes fields such as name for the sub-service name and template for associating specific user interface templates.

Figure 11 provides an illustrative example of the schema. Consider an application dedicated to smart wristbands, in which one of the key e-services is “User Monitoring.” In this e-service, various sub-services are defined to encapsulate specific features and functionalities. The example shows the recommendation engine returning recommended sub-services early in the morning for a user: “sleep monitor,” “heart rate monitor,” and “exercise monitor.”

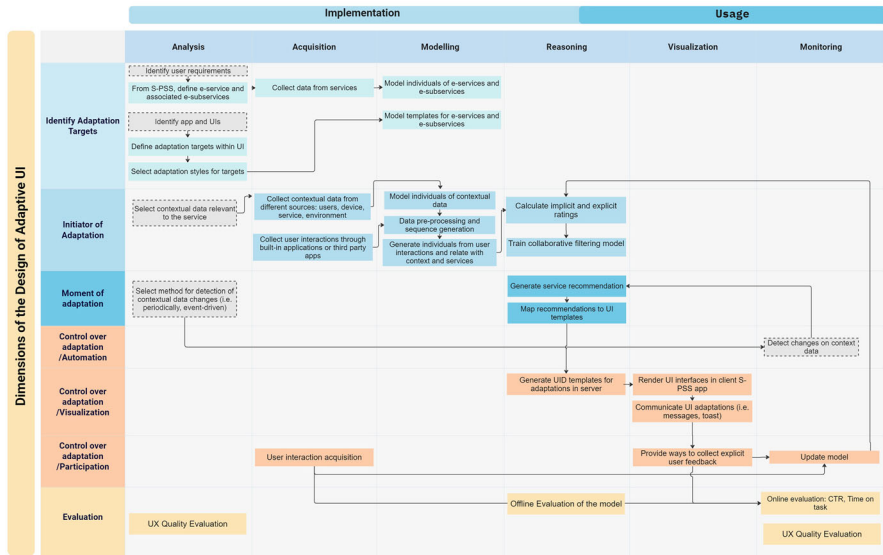


Fig. 12 Roadmap for implementation incorporating design and implementation dimensions

The query method “mapRecommendedUITemplate” takes recommended service and sub-service as parameters in JSON format and navigates the hierarchy of e-services and e-sub-services to identify the associated template and map the UI elements. The algorithm for this process is presented in Appendix 3.2 Algorithm 1; it is important to note that a template for a service will contain one section which will hold the recommended sub-services (Line 6). This subsection then forms the parent from the sub-services templates; each template can have their own root container or use the root container from the parent (Line 13). The method result is a template in a JSON format, as shown in Fig. 11.

### 3.5 A roadmap for framework implementation

A roadmap for implementation is presented in the form of a matrix as depicted in Fig. 12. The matrix establishes a connection between the dimensions essential for designing an AUI for S-PSS outlined in Sect. 3.4.1, which are represented as rows. The columns correspond to various phases or stages of the framework, as detailed in Sect. 3. The roadmap is a visual representation of the framework progression and aligns with the principles of User Centred Design (UCD), encompassing both explicit and implicit participation. Implicit participation is facilitated via user interaction data, while explicit participation is encouraged through explicit methods for rating services and the use of questionnaires for UX quality evaluation. These mechanisms contribute to the generation of adaptive UI tailored for S-PSS.

One critical stage for the framework implementation, as specified by the UCD, is based on understanding and specifying requirements and usage contexts. In the roadmap presented in Fig. 12, this pre-implementation stage is included, referred to

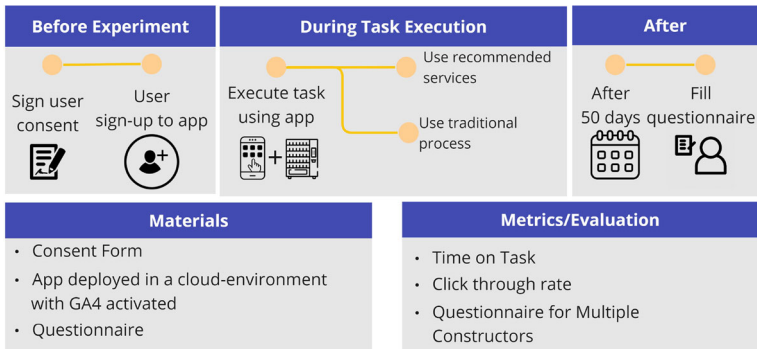


Fig. 13 Protocol case study

as “Analysis.” Some of the steps of this stage are represented in gray, which indicates that any method used is considered outside the scope of the framework, but must be taken into account to ensure correct implementation.

## 4 Case study

The proposed framework and each of its elements were validated through a case study on a smart coffee vending machine, highlighting the increasing interest of major international manufacturers in smart appliances, due to their significant market growth potential (Yu and Sung 2023). The case study focuses on analyzing user interaction data from a SCP and proposes an adaptation of the UI in a web app. This adaptation takes into account the context and previous interactions of the users, aiming to enhance their overall experience with the machine.

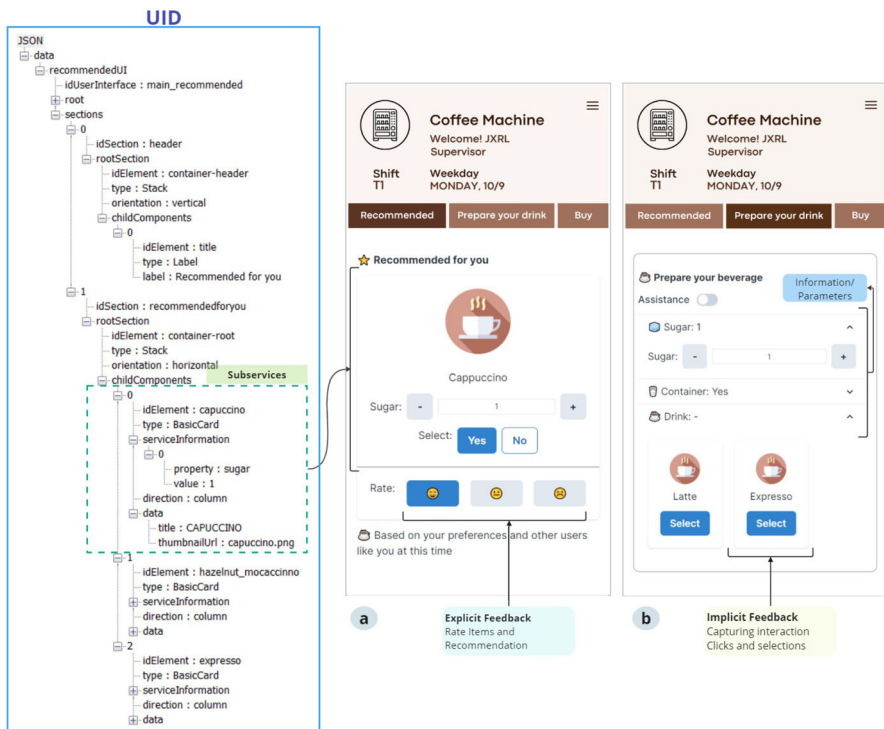
### 4.1 Case study setup

The protocol followed for the case study is presented in Fig. 13. For this study, a responsive web app was developed for the machine using the Next.JS framework and Chakra UI. Interactions of 45 users aged between 19 and 40+ years were used and collected in this experiment over a period of 50 days (see Table 6). These users registered on the application by providing a username and selecting a role for themselves. Prior to participating, each user formally consented to have their interactions recorded through the application. In total, 10,608 interactions are used. All data were anonymized and it does not contain sensitive user information.

In the process of generating the UI for the smart vending machine, the adaptation target was the “Recommended for you” e-service, and the bundle of sub-services represented the types of beverages available from the machine. This information was registered in the ontology prior to the experiment. In total, 18 possible services were registered.

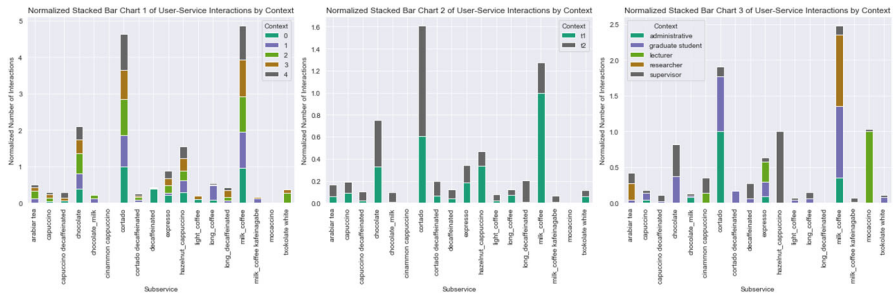
**Table 6** Participants demographics

Attribute	Item	Freq	Attribute	Item	Freq	Attribute	Item	Freq
<b>Age</b>	19–25	2	<b>Gender</b>	Male	36	<b>Occupation</b>	Graduate student	11
	26–30	25		Female	9		Supervisor	12
	31–35	7		Non-Specified	0		Administrative	5
	36–40	8			Researcher		9	
	>40	3			Lecturer		8	
Total users	45							



**Fig. 14** User Interface smart product application

Figure 14 presents part of the UI utilized in the experiment, featuring the initial screen titled the “Recommended for you” tab. This screen is dynamically generated based on user-specific contextual data, updating accordingly if these values change. Employing the adaptation style of folding irrelevant items and prioritizing the top three recommended services, the UI presents these recommendations in the form of card components, which are gaining popularity in many mobile and web apps (Rodrigues et al. 2017). These cards are constructed based on a User Interface Descriptor provided in JSON format. This descriptor is generated using input from the recommendation engine, as well as information derived from the services specified in the ontology and



**Fig. 15** Contextual data stacked bar representation (1) Weekday, (2) Shift, (3) role

the related information they require to function effectively. Table 7 breaks down the analysis of the rest of design dimensions for UI adaptation in this case study.

Users had also the choice to complete the process on the second tab (*Prepare your beverage* tab), if recommended services were not good selections. The UI allowed the users to rate services recommended and in this way gather explicit feedback from users (see Fig. 14).

## 4.2 Technical implementation overview

For the data acquisition, user interactions within the app were captured using Google Analytics 4 (GA4) and stored in BigQuery. GA4 is a widely used web analytics service that effectively tracks and reports website traffic and user interactions. These captured interactions were then pre-processed and uploaded in batches using the “Data Importer” component of the framework. The information was subsequently stored in GraphDB, a graph database that adheres to the RDF and SPARQL standards. Out of the total 10,608 individual interactions recorded, certain events such as “page views” or “first visit,” which were deemed irrelevant to the study, were filtered out and discarded. Each element of the UI was given a unique identifier, and only interactions involving these elements were considered to generate a subset of 1,458 valid interaction sequences. On average, each sequence consisted of seven interactions occurring within a user session, encompassing all relevant actions related to the service. These actions include clicks on buttons and tabs, as well as session initiation. Table 8 presents an example of a valid sequence.

Three types of contextual data are considered: *user role*, *shift*, *weekday* that represent user and time as presented in Fig. 15 which illustrate the analysis of contextual data through a stacked bar chart. Each bar in the chart is segmented to represent the distribution of interactions across different categories of these contextual variables. While it is expected that certain services receive more interactions, the chart also highlights instances of higher interaction volumes within specific contexts.

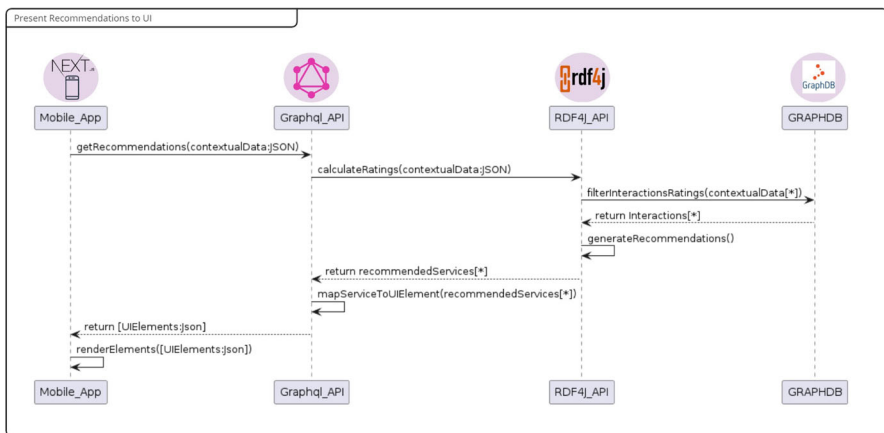
To illustrate the flow of events involved in creating the AUI with service recommendations, a sequence diagram (Fig. 16) is presented. The UI was developed using Next.js and React to create UI elements and layout. In this particular case, the triggering events for generating recommendations are “user login” or changes in contextual

**Table 7** Dimensions considered in the case study

Dimensions	Application in case study						
Adaptation target	Mobile App Presentation of services and suggested parameters through the modification of User Interface Descriptors.						
Initiator of the adaptation	Analysis of user interactions through User-based collaborative filtering.						
Control over the adaptation	<table border="1"> <thead> <tr> <th data-bbox="526 573 585 790">Recommendation-based Automation</th> <th data-bbox="526 790 585 1007">Participation</th> <th data-bbox="526 1007 585 1201">Visibility</th> </tr> </thead> <tbody> <tr> <td data-bbox="597 573 656 790">System automatically identifies changes in context</td> <td data-bbox="597 790 656 1007">User can accept or decline recommendation</td> <td data-bbox="597 1007 656 1201">Cards with recommended services are shown to user. Folding irrelevant services and reorder services according to recommendations</td> </tr> </tbody> </table>	Recommendation-based Automation	Participation	Visibility	System automatically identifies changes in context	User can accept or decline recommendation	Cards with recommended services are shown to user. Folding irrelevant services and reorder services according to recommendations
Recommendation-based Automation	Participation	Visibility					
System automatically identifies changes in context	User can accept or decline recommendation	Cards with recommended services are shown to user. Folding irrelevant services and reorder services according to recommendations					
Moment of the adaptation	Changes in contextual data: Shift, User Role, Service availability						
Evaluation of the adaptation	Feedback based: Survey with participants. Analytic based: CTR, Time on task.						

**Table 8** Example of interaction sequence collected by the system

Event timestamp	User id	Event label	Event data
1678879326812650	OCUnH	Btn5meh_capuccino_click	Service:capuccino, information: [parameter:sugar, value:0]
1678879326812650	OCUnH	Tabpreparation_click	Preparation
1678879332853910	OCUnH	Btn1sugar_click	–
1678879341145730	OCUnH	Btn3sugardec_click	–
1678879346246940	OCUnH	Btn1drink_click	–
1678879346246940	OCUnH	Btn5expresso_click	Service:expresso, information: [parameter: sugar,value: 0], {parameter: cup,value: Yes}}
1678879413245150	OCUnH	Btn1ok_click	Buy

**Fig. 16** Sequence diagram illustrating the process of generating recommendations for a mobile app, involving the Mobile App, APIs, RDF database, and UI rendering

data detected on the web app. To ensure a clear separation between the front-end and RS component, the framework employs separate connector APIs. This design choice allows the recommendations system to remain decoupled from specific UI implementations, promoting flexibility and enabling usage across various scenarios and with different UIs.

The front-end API employs GraphQL,<sup>2</sup> a powerful query language that employs a strongly typed schema to define available data and operations. The UI schema<sup>3</sup> created acts as a User Interface Descriptor that represents the UI elements that will be rendered on the user application's interface. GraphQL connects to the Recommendation System API, which was developed using RDF4J, a robust Java-based framework for working

<sup>2</sup> <https://graphql.org/>.

<sup>3</sup> <https://mu-sse.github.io/adaptiveUIs-project/app-vending-machine/server-backenddrivenUI/src/schema/types.ts>



with RDF data. The RS API<sup>4</sup> encompasses a collection of operations that uses SPARQL statements. These include batch operations for populating the ontology, retrieving and filtering information from it, and executing tasks pertinent to the CARS model.

When service recommendations are retrieved, they are returned to GraphQL API. These recommendations, based on the obtained data, are transformed into UI elements using the descriptor template created for each service. The back-end then constructs a response object in JSON format and sends it back to the front-end as presented in Fig. 14. Upon receiving the response, the front-end renders the UI elements accordingly, presenting the recommended services to the user. In the Next.js framework, these elements are implemented as self-contained snippets of code known as components.

The JSON response depicted in Fig. 14 presents a visual representation of the GraphQL schema definition, utilized for the creation of the AUI for recommending services. A *Card* is a self-contained component that can display information but also contain other elements. This allows for multiple cards to be displayed in a horizontal arrangement. The *'recommendedServices'* query is defined to fetch a set of recommended items based on user-specific parameters, returning an *'Stack'* with the recommended cards. The schema previously defined on section 3.4.2 is standardized description for the UI elements and allows the server to generate, modify, and send the components to the client side for rendering. The client side, developed in Next.JS has an implementation of these components so it can interpret them and generate the appropriate visual elements and layout. Additionally, the client side can handle user interactions and communicate them back to the server for processing.

This approach ensures that by implementing the UI elements on the client side, the system can consistently render and utilize the same schema across multiple platforms, promoting reusability, extensibility, and ease of development.

### 4.3 Evaluation

The evaluation of the framework implementation is presented in the following subsections. Firstly, the recommendation component is assessed through offline evaluation using a dataset derived from user interactions. Subsequently, the evaluation of UX is conducted, incorporating both quantitative and qualitative data obtained from a questionnaire. Finally, the online evaluation derives from real-time metrics collected during actual usage.

#### 4.3.1 Recommendation component evaluation

The effectiveness of the recommendation component in the framework was evaluated through an offline assessment using the collected dataset (Herlocker et al. 2004). The evaluation comprised a series of experiments comparing the Precision@k scores and Precision@k scores and Mean Average Error (MAE). Precision@k is a widely-used metric in Recommendation Systems, quantifying the accuracy of the top k recommended items for a user by computing the ratio of relevant items (i.e., those the user interacted with or liked in the test set) among the top k recommendations. On the

---

<sup>4</sup> <https://github.com/mu-sse/adaptiveUIs-project/tree/main/contextawareUX4S-PSS>

other hand, MAE, another popular metric, measures rating prediction performance by calculating the average absolute difference between predicted and actual user ratings in the test set (Herlocker et al. 2004).

As described in section 3.3, the interactions captured were treated as implicit feedback. For explicit feedback, in the app, users had the option to express their opinion on the recommended items by rating them using emoticons. The emoticons consisted of sad, indifferent, or happy faces, which corresponded to a scale from 1 to 3. However, it is important to note that users were not obliged to provide this feedback, and these ratings represent only a  $\approx 7\%$  of the total number of interactions.

In S-PSS, to provide service recommendations to users a straightforward assumption is that users will prefer services they have recently selected or rated, since familiarity reinforces trust in recommendation. This *naive* approach is easy to implement and leads to the presumption that users' behavior could be relatively constant when using smart products, and it was used to obtain and present the information required for the service (e.g., sugar level).

However, to recommend services, this approach has some limitations as it does not take into account users' individual context, which can vary over time, leading to poor recommendation quality. Nonetheless, this approach can be useful as a baseline for evaluating the performance of more specialized recommendation systems in S-PSS. This naive approach reported a Precision@1 of 0.559, indicating that this simple method may have some effectiveness in recommending services to users.

Each set of experiments was repeated 20 times using different randomly selected training and test sets each time (70–30 ratio, respectively); then, the average values were taken for each of the metrics. The first experiment compares the Precision@k without using contextual data with a kNN model using Pearson correlation factor, and cosine, and Euclidean distance metrics. The Precision@K scores were evaluated at  $k=1, 2,$  and  $3$  as demonstrated in Fig. 17.

On average, the kNN with cosine distance performed slightly better than Pearson. There were some noticeable differences in the effectiveness with Euclidean distance at  $k=2$ , the median score was 0.57, and  $k=3$  the median score was 0.51. This could indicate that cosine distance and Pearson correlation factor are more suitable in this scenario. Unlike Euclidean distance, the Pearson correlation coefficient takes into account the mean and standard deviation of the ratings, which makes it more robust to variations in the magnitude of ratings (Ricci et al. 2011).

There were a few outliers in the boxplots, particularly for kNN with cosine distance at  $k=3$ . These outliers may indicate cases where the model was ineffective for certain users and services.

The interquartile ranges for all the boxplots indicate a relatively normal distribution in the majority, indicating that the Precision@k scores were generally consistent across different users and services.

When we compared the Precision@k scores for different methods to the baseline precision score of the naive method, it is demonstrated that the kNN models outperformed the baseline, with kNN with cosine distance performing in a more constant way overall.

The results of Precision@k using contextual data are presented in Fig. 18, and in Table 9, which represent an improvement with the use of contextual data, the behavior

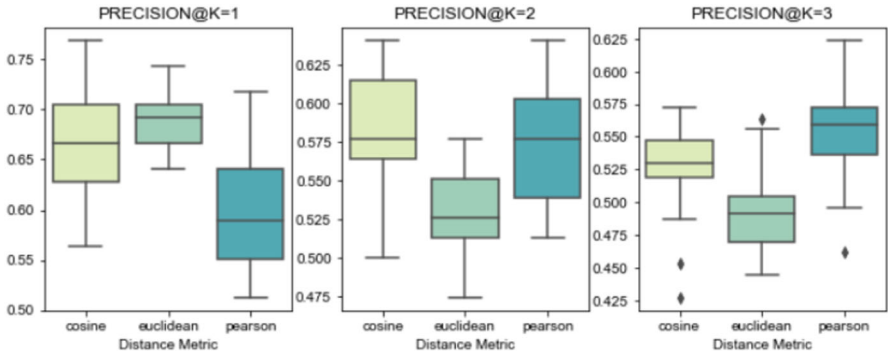


Fig. 17 Results Precision@K without using contextual data

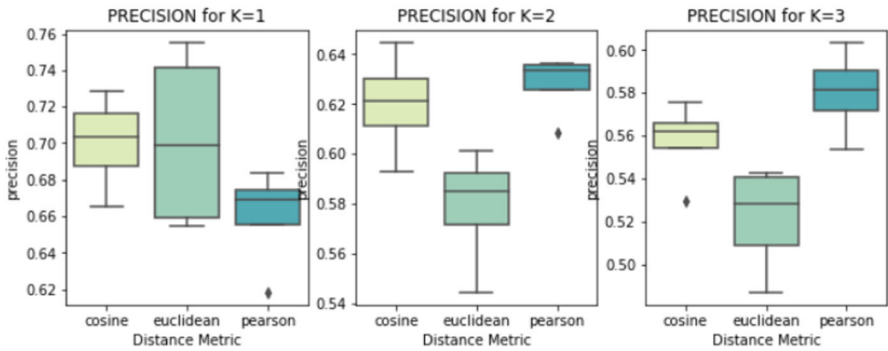


Fig. 18 Results Precision@K with context

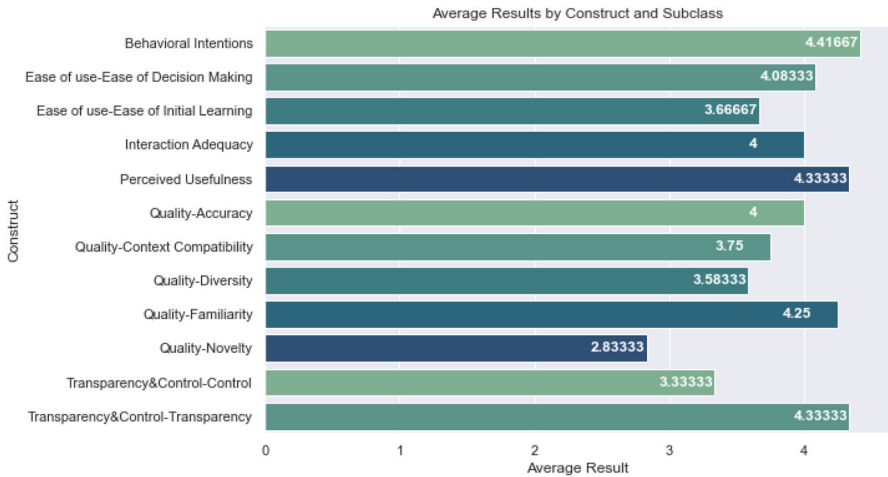
among distance metrics is similar without contextual data, which is expected. At Precision@1, on average Euclidean and cosine distances perform better. This can be due to effectively identifying the closest item to the user’s input, based on the shortest distance in the feature space. However, as  $k$  increases, Euclidean distance is outperformed by cosine and Pearson, possibly because does not take into account the correlations among users beyond their pairwise distance (Ricci et al. 2011). On the other hand, Pearson slightly improves its performance among the other distance metrics as  $k$  increases which could be due to the sensitivity of Pearson correlation to outliers, as the algorithm has the opportunity to recover from one bad recommendation at Precision@1 and still deliver a more accurate result.

In addition, Table 9 presents the average MAE results for the two experiments. A lower MAE indicates a better performance of the model in predicting rating values. The results show a slight improvement in the second test. This suggests that the model can predict better the rating values in the second test using contextual data.

After evaluation, we considered the use of the cosine distance metric as it outperformed Euclidean distance for higher values of  $k$  in terms of Precision and Pearson at  $k=1$ . By considering the correlations among users beyond pairwise distance, cosine distance provides a better understanding of user preferences and improves the precision of recommendations.

**Table 9** Results of Precision@K and MAE

	No context			With contextual data		
	Cosine	Euclidean	Pearson	Cosine	Euclidean	Pearson
Precision@1	0.664957	0.688889	0.600000	0.700264	0.702035	0.660094
Precision@2	0.582692	0.528846	0.573077	0.622402	0.575426	0.627981
Precision@3	0.527350	0.491880	0.556410	0.556113	0.517175	0.587105
MAE	0.114175	0.132899	0.101554	0.109843	0.122772	0.094235

**Fig. 19** Results of questionnaire by constructs and subclassification

### 4.3.2 Validation of the UX

To assess the UX of the framework implementation, participants provided their feedback through a questionnaire. The questionnaire was developed based on the user-centric evaluation framework proposed by Pu et al. (2011), specifically tailored to the context of Recommendation Systems. Questions were developed using the pre-defined constructs provided in the framework, and the questions are related to these constructs and adapted to ensure relevance to the S-PSS context.

The primary objective of this questionnaire was to gather valuable user feedback from participants (see participants in Table 6) and evaluate several aspects of their experience, including behavioral intentions, ease of use, perceived quality, beliefs, and, attitudes, toward the S-PSS. To capture user responses effectively, a 5-point Likert scale ranging from “strongly disagree” (1) to “strongly agree” (5) was employed for each question. Figure 19 presents the results for each construct and subclassification when necessary.

The questionnaire and its results, presented in Table 10, provide valuable insights into various aspects of the UX. The behavioral intentions were measured in terms of the user’s intention to continue using the app and their intention to return, users presented a strong willingness to keep using a smart product like the one presented. To assess the ease of use, two key aspects were examined: ease of decision-making

**Table 10** Case study: UX constructs and survey results

Construct	Subclass	Question	Mean	SD
Behavioral intentions	-	I will use a smart vending coffee machine like this to get my product	4.417	0.669
Ease of use	Ease of decision-making	Using the services recommended to me saved me time	4.083	0.669
	Ease of initial learning	If it were my first time using a smart coffee vending machine, the app would facilitate use (instructional support)	3.667	1.155
Interaction adequacy	Interaction adequacy	I easily found the recommended services	4.167	1.115
		The app provides an appropriate way for me to express my preferences	3.833	1.337
Perceived Usefulness	-	. I feel supported to find an appropriate service for me with the app	4.333	0.778
Quality	Accuracy	Overall, the services recommended matched my preferences	4.000	0.953
	Context compatibility	The services recommended took my personal context requirements into consideration	3.750	0.866
Transparency & Control	Diversity	The services recommended are diverse	3.583	0.900
	Familiarity	I am familiar with the services recommended	4.250	0.622
	Novelty	The app helped me to discover new services that I might like	2.833	1.193
	Control	The app seems to control my decision process rather than me	3.333	1.669
	Transparency	Do you agree with the following statement? As a user, I understand why certain services are being recommended	4.333	0.651

and ease of initial learning. Ease of decision-making refers to the users' ability to make choices effortlessly within the app, while ease of initial learning focuses on their understanding of how the app functions, how to provide input, and how their preferences are incorporated. For instance, one user expressed their desire to keep using the app, but they were concerned that it might not fully understand their preferences after just a single use. This highlights the importance of the app's ability to quickly grasp and adapt to the user's individual needs. Interaction adequacy was assessed to determine if users felt that the app provided an appropriate way for them to express their preferences. Perceived usefulness, on the other hand, refers to the extent in which users felt that using the recommended services would enhance their performance as a support mechanism.

When it comes to quality, one aspect evaluated was the accuracy of the recommendations in matching the user's preferences. It was found that users generally felt that the overall services recommended aligned well with their preferences. Context compatibility examined whether the system understood the users' personal context requirements, and it was determined that the services recommended took these requirements into consideration to a satisfactory degree. Familiarity explored whether users were already familiar with the services recommended to them. In this case, users strongly agreed that they were familiar with the suggested services.

Novelty assessed the extent to which users received recommendations for new types of services that they want to accept or test. In our implementation, some users felt that the app did not assist them in finding anything new, which can jeopardize user overall satisfaction depending on the scenario, where users can be frustrated by an excessive familiarity and lack of novelty and diversity (i.e., music, streaming services). However, for the context of the case study in creating Adaptive User Interfaces, there were no restrictions on suggesting services that users had already experienced in the past, which likely contributed to the users' strong familiarity with the services recommended. One user expressed that the absence of images made it difficult to be persuaded to explore options beyond what they already knew. This highlights the importance of User Interface design and visual elements. Diversity measured the level of diversity in the recommendation list, and users indicated that the services recommended to them exhibited a moderate level of diversity.

Lastly, transparency, and control were evaluated. User control assessed whether users felt in control during their interaction with the app. Users provided a neutral response, suggesting that they did not feel completely in control of the decision-making process. This could be attributed to a lack of understanding on how to reject a recommendation, as this functionality was not provided in the interface. However, users did have the ability to rate recommendations, although not all users exercised this option. Regarding transparency, users strongly agreed that they understood why certain services were being recommended to them. This suggests that the app effectively conveyed the reasoning behind its recommendations. Further analysis and discussion of these findings will be presented in the subsequent section.

In addition to the questionnaire, an online evaluation was conducted to measure metrics relevant to the recommendation component performance and effectiveness (Herlocker et al. 2004). One of these metrics was "Time on task" which served as an indicator of the app's ease of use. The time taken by users that did not use the services

**Table 11** Usability metrics

Metric	Result
CTR = $\frac{\text{Total number of acceptances}}{\text{Total number of sequences}}$	0.6913
Avg. time on task = $\frac{TT_1+TT_2+\dots+TT_n}{n}$	00:49 s (accepted) 01:44 s (rejected)
Online evaluation	

recommended was compared with those who did. This allowed for a comparison of the app's usability improvements. Another metric considered was the Click-Through Rate (CTR), which measures the effectiveness of service recommendations. It quantifies how often the recommended services were successfully accepted by users, indicating their relevance and usefulness without considering the ranking of the services presented (the user was given only the top 3 options). As presented in Table 11, CTR is 69.13%, and the average time on task was quantified in seconds for those users that use the "Recommended for you" tab and those who did the process manually. When it comes to the average time on task users that use recommended services completed the task 55% faster.

## 5 Discussion

In this section, we will discuss the findings and limitations of the results. The study aimed to investigate the impact of implementing a proposed framework for adaptive user experiences in S-PSS.

### 5.1 Impact on performance and usability

When analyzing performance and usability, these can be related to efficiency and *ease of use* (Pu et al. 2011). On the questionnaire, users strongly agreed that the use of the "recommended for you" tab on the app saved them time (Fig. 19). This feature, which is the result of adaptation and personalization, is a helpful decision-making tool, resulting in users feeling more confident in their choices. The *perceived usefulness* of this feature is an indicator of the extent to which users feel a performance improvement in the system's usage or in their sense of competence. Importantly, the results reveal a strong positive correlation ( $r = 0.64$ ) between users' perceived usefulness and the ease of decision-making (Fig. 20).

This ease of use can be considered a form of efficiency, as it reduces the cognitive load and effort required for decision-making. This observation is supported by the data collected, which shows a decrease in the time spent on tasks compared to performing them manually. Combining task completion time with user feedback provides a more comprehensive understanding of user behavior. These results emphasize the importance of a User Interface that supports users in finding suitable services quickly and conveniently.



**Fig. 20** Correlation matrix UX constructs in RS

Additionally, *interaction adequacy* played a crucial role, as users expressed that the recommended services in the app were easy to find and it provided an adequate way for them to express their preferences. This implies that a well-designed and intuitive interface contributes to a positive UX and enhances usability. Nonetheless, it is important to note that *interaction adequacy* exhibits a strong correlation with the *accuracy* and *context compatibility*. This clearly indicates that the effectiveness of finding the recommended services is strongly influenced by the degree to which the interface aligns with the user's needs and context.

In terms of the *ease of initial learning*, there seems to be a slight decrease. While users perceive the adaptation of the interface as a time-saving tool, it may not necessarily provide strong instructional support during the initial stages of use or improve the learnability of the system. This suggests that users may require some time to familiarize themselves with the digital services provided by the app and understand how to effectively leverage its recommendations. Further attention could be given to improving the instructional and learnability aspects, particularly during the initial stages of user interaction with the smart products and the service system, when it is not clear to them how recommendations are generated.



## 5.2 Impact on user engagement and satisfaction in S-PSS

The observations made in the study provide valuable insights into the impact of the proposed framework on user satisfaction and engagement in S-PSS.

Engagement refers to the level of user involvement with a particular entity. In this context, one commonly used metric to measure engagement is the CTR. In the results, 69.13% of the time the recommendations were used. It is important to note that CTR is not necessarily correlated with metrics on the offline evaluation and can potentially increase as users interact more with the system and feed the RS. However, this scenario can vary and is highly dependent on the complexity of the system being evaluated. In this case, the experiment focuses on a relatively simple product, which suggests that the impact on engagement might be more straightforward to analyze. Nonetheless, the experiment itself had other constraints to tackle such as time limitations from users. Despite these constraints, conducting the experiment remains valuable as it provides valuable insights into user behavior and system performance.

Analyzing the specific questionnaire results, it is evident that several factors can influence user engagement. The aspects related to the quality of recommendations, such as accuracy, familiarity, and context compatibility, all received positive feedback from the users. Furthermore, their perception of accuracy is positively correlated with context compatibility of services being delivered on the interface as presented in Fig. 20

This indicates that when the recommended services align with users' preferences and consider their personal context requirements, it positively impacts their engagement. This is also backed-up with the CTR that highlights that more than half of the times user preferred the interface personalized to them.

On the other hand, user satisfaction can be influenced by many factors and can not solely be attributed with high recommendation accuracy (Kaminskas and Bridge 2016). The results from the study further support this notion, as presented in Fig. 20, where the *behavioral intentions* aimed to assess users' willingness to continue using the presented smart product, which implies a level of satisfaction and their desire to return to the product. This construct reveals a weaker correlation with *accuracy* and *context compatibility* and highlights that user satisfaction is closely related to the ease of use, in this particular case with ease of decision-making and perceived usefulness. Nevertheless, accuracy and context compatibility remain crucial factors that positively contribute to these constructs. This highlights the importance of considering the system as a whole, where a user-friendly interface and seamless user experience contribute to satisfaction, alongside accurate and compatible adaptations for User Interfaces.

## 5.3 Findings and limitations

- *Novelty and Familiarity on S-PSS UX:*

Users exhibit a preference for recommendations of previously experienced items, as Kaminskas and Bridge (2016) highlighted “familiarity” can reinforce trust in a system. The questionnaire revealed that users strongly agreed that the recommended services were familiar to them. However, they expressed disagreement regarding the novelty aspect, indicating that they did not discover new services

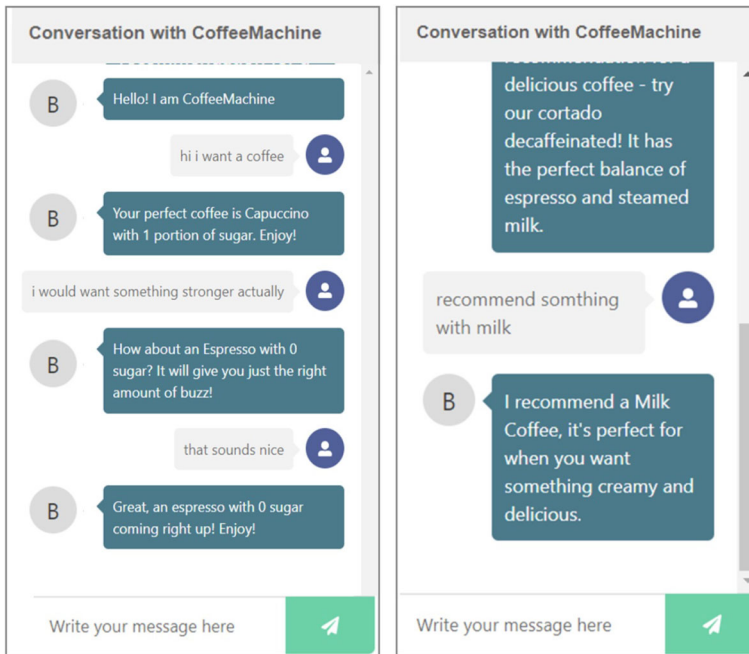


Fig. 21 Integration of recommendation component and chatbot

beyond their existing preferences (2.8/5 score). Interestingly, this lack of novelty did not impact their intention to continue using a product of this nature in the future. To validate whether this behavior extends to other types of smart products, further studies are necessary.

Additionally, the study found that *Diversity* of recommendations fell within the middle range (3.5/5 score), suggesting that users perceived a certain lack of diversity among the recommended services. This perceived lack of diversity did not affect their intention to use the system. Despite this, it is important to address the balance between familiarity and diversity in the recommendations. The recommendation component developed under the kNN algorithm looks for similar users and rewards more interacted services within that neighborhood or group. While users did find some diversity in services interacted with by similar users, they did not necessarily perceive it as novel. Therefore, to balance out familiarity, diversity and novelty, it is necessary to consider the inclusion of serendipitous recommendations that are unexpected but still can be relevant (Ziarani and Ravanmehr 2021). Including serendipity into the recommendation process can encourage exploration and promote less known services, and it is an aspect that should be further analyzed on the context of the generation of Adaptive User Interfaces and smart products. These limitations highlight the need for additional research to explore how users' preferences for familiarity and the perceived diversity of recommendations may vary across different smart products. Understanding these dynamics will provide

insights for improving recommendation algorithms and tailoring them to meet user expectations and preferences effectively.

- *Technological Limitations*

The comparison of several collaborative filtering algorithms in the given context could be further studied. However, it is necessary to align this exploration with the study of users' preferences when using digital products as mentioned in previous point. In the conducted experiment, a well-known machine learning algorithm was employed (kNN), which had been tested and analyzed in multiple scenarios. Nonetheless, it is important to understand when it is necessary to apply more robust or resource-consuming machine learning algorithms. It is important to weigh the trade-offs between algorithmic performance and user requirements. Even when more complex and computationally expensive algorithms may offer improved accuracy or the ability to handle larger datasets, they might come at the cost of increased computational resources, longer training times, or reduced scalability. Therefore, it is important to note the diversity of user preferences. Future research can focus into specific scenarios where the benefits of employing resource-intensive algorithms outweigh these trade-offs. Additionally, it was observed that several users missed the integration of recommendations into the service parametrization, which refers to the necessary information required for a service to function. In the app, this integration was achieved using a naive approach that only presented the most recently used information. Users expressed that the app did not learn their specific preferences on the services, such as the amount of sugar they prefer in particular beverages. This aspect could be more complex in different scenarios, and the parametrization of services should be considered in future work as well as part of the adaptation.

- *Interface Design Considerations:*

In the context of adaptation, the elements of design play an important part. It is essential to consider not only the content and technical implementation but also the structure and visual presentation of the interface (*interface adequacy*) (Ozok et al. 2010). While this study primarily focused on measuring *interaction adequacy*, valuable user feedback highlighted issues such as the difficulty of reading pop-up messages in the app and the lack of visual elements. This feedback emphasizes the significance of addressing interface design aspects in future work. Although this study did not focus on the specific design elements or the optimal design structure for adaptation, it is still an important consideration. The study mainly focused on the technical and practical aspects of the implementation. Further research should explore how the interface can be optimized to enhance readability, clarity, and overall user experience on S-PSS.

- *Integration with Multi-modal User Interfaces*

The integration of the framework with multi-modal UI currently falls outside the scope of our framework. Multi-modal UI enables users to engage with the machine using various input and output methods, including touch, voice, text, gestures, or mouse. These hybrid interfaces, particularly those incorporating natural modes of interaction such as conversations, are quickly becoming a new standard. Therefore, understanding how to integrate conversational instructions with those using a mouse or gestures is crucial, as the use of mobile and web applications is unlikely

to disappear entirely. The coexistence and integration with new technologies are essential for generating adaptations.

The expansion of conversational User Interfaces has witnessed remarkable growth in recent years (Haugeland et al. 2022), exploiting techniques such as natural language processing and Large Language Models (LLM). Incorporating these advancements into the UI can enhance user engagement and satisfaction. By enabling users to communicate with the Recommendation System through natural language commands, S-PSS can deliver a more personalized and user-friendly experience, catering to evolving user expectations and preferences.

The decoupling of the Recommendation System and AUI generation components in the framework design allows each component to evolve independently, easing adaptation to different UI platforms without affecting the core Recommendation System functionality.

As a way to exemplify the diversity of UIs within an S-PSS environment, a chatbot was developed using Next.js and integrating the Open AI API<sup>5</sup> for natural language processing, which includes large language processing capabilities. This chatbot serves as an additional interface that seamlessly integrates the Recommendation System component and the underlying ontology. The chatbot is capable of generating recommendations in a conversational manner. As presented in Fig. 21 the user asks for the service and the recommendation is presented in a more natural and conversational style. In the event that users are unsatisfied with the provided option, they have the ability to request changes or modifications through natural language commands. The Open AI API receives commands as inputs. In this case, the command includes the response of the recommendation API which is an ordered set of recommended options in JSON format. The language model underlying the chatbot considers these commands and the available options provided by the Recommendation System to generate an appropriate response, ensuring a more interactive and user-centric experience within the S-PSS framework. This can provide a more natural way for users to access S-PSS digital services. However, the current implementation of the Recommendation System primarily relies on user interactions. In future research, it would be valuable to explore the integration of a natural User Interface that can recognize positive or negative implicit feedback on conversational manner and integrating with user interactions as well. This would require further investigation and development to enhance the system's ability to capture user preferences and improve the overall recommendation process. Furthermore, it is necessary to analyze the UX impact and how it would compare with traditional graphic interfaces.

However, these data-driven approaches will necessitate access to users personal and behavioral data, which can vary depending on the application domains of S-PSS and the type of UI. It is imperative that the services created enable users to make informed decisions about their privacy. Users should also have control over their personal information (van Ooijen and Vrabc 2019). Adhering to privacy regulations such as the General Data Protection Regulation (GDPR), implemented

<sup>5</sup> <https://platform.openai.com/docs/guides/chat>

by the European Union to regulate user privacy, can help ensure users consent and awareness of the data collected (Zheng et al. 2019).

## 6 Conclusion

This paper presented a development approach for recommendation-based Adaptive User Interfaces, within the S-PSS environment. A framework was used as a structure to guide the system development and aimed to exploit user interactions with digital services and contextual data. One of the main contributions of this work lies on the integration of various components to achieve the creation of AUI for digital services. To demonstrate the practical implementation, a case study was presented with a hands-on development approach, considering technological aspects and utilizing appropriate tools. The results demonstrated an improvement on the precision of the recommendations using a context-aware approach. Furthermore, the framework presented a decoupled approach for the generation of the User Interfaces.

Regarding the UX, the findings and results examined various aspects, such as, user performance, usability, engagement, and satisfaction. The implementation of the framework demonstrated positive effects on user efficiency and ease of decision-making. The User Interface generated using the framework saved users' time and increased their confidence in their choices. However, there were slight differences in terms of ease of initial learning, indicating that users may still require time to familiarize themselves with the digital services and understand how recommendations are generated, to increase their trust.

User engagement and usability were strongly influenced by the quality of recommendations, including accuracy, and context compatibility. The study revealed that when recommended services aligned with users' preferences, and considered their personal context requirements, it positively impacted their perceived usefulness. Interaction adequacy, expressed through an intuitive interface that allowed users to express their preferences and easily find recommended services, also played a crucial role in enhancing user satisfaction and usability. The ease of use aspect was reflected in users' feedback, emphasizing the importance of a streamlined User Interface and a system that supports quick and convenient service discovery.

While the study provided valuable insights, there were limitations and areas for future research. Users expressed a preference for familiarity in recommendations and perceived a lack of novelty and diversity, which calls for further exploration of users' preferences across different smart products. Technological limitations were also discussed, highlighting the need to further study algorithmic performance and resource requirements. Additionally, interface design considerations, such as readability, clarity, and visual elements, emerged as important factors for optimizing UX in S-PSS. Furthermore, the expansion of natural User Interfaces and conversational Recommendation Systems has witnessed remarkable growth in recent years. This trend highlights the importance of considering these interfaces within the S-PSS environment to provide several means of interaction.

In conclusion, this study contributes to understanding the impact of adaptive user experiences in S-PSS, emphasizing the significance of efficient decision-making,

user-friendly interfaces, and personalized recommendations. Future research should address the limitations identified and further explore user preferences, algorithmic approaches, and interface design considerations to enhance the effectiveness and user satisfaction in S-PSS.

## Appendix

---

### Algorithm 1 Map Recommended Services to UI Template

---

```

1: function MAPRECOMMENDEDUITEMPLATE(recommendedService)
2:   // Get the service template based on the recommended service
3:   serviceTemplate ← GETSERVICETEMPLATE(recommendedService.service)
4:
5:   // Get the parent section (sub-service container) from the service
   template
6:   parentSection ← GETSUBSERVICECONTAINER(serviceTemplate.sections)
7:
8:   for each sub-service in recommendedService.subservices do
9:     subserviceTemplate ← GETSUBSERVICETEMPLATE(subservice)
10:    // Iterate over sections in the sub-service template
11:    for each childSection in subserviceTemplate.sections do
12:      for each element in childSection.uiElements do
13:        targetContainer ← childSection.rootSection ? childSection.rootSection :
14:        parentSection.rootSection
15:        // Add the element to the target container
16:        targetContainer.childComponents.push(element)
17:      end for
18:      // If the child section has a root, add it to the root parent
19:      if childSection.rootSection then
20:        parentSection.rootSection.childComponents.push(childSection.rootSection)
21:      end if
22:    end for
23:  end for
24:  // Return the modified service template
25:  return serviceTemplate
26: end function

```

---

**Acknowledgements** This project is funded by the European Union’s Horizon 2020 program (Marie Skłodowska-Curie Grant No. 814078).

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: International symposium on handheld and ubiquitous computing, pp. 304–307. Springer (1999)
- Abrahão, S., Insfran, E., Sluÿters, A., Vanderdonckt, J.: Model-based intelligent user interface adaptation: challenges and future directions. *Softw. Syst. Model.* **20**(5), 1335–1349 (2021)
- Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems, Recommender systems handbook, 217–253. Springer (2010)
- Ali, M., Khan, S.U.R., Mashkoor, A., Taskeen, A.: A conceptual framework for context-driven self-adaptive intelligent user interface based on android. *Cognit. Technol. Work* **26**(1), 83–106 (2024)
- Bailly, G., Lecolinet, E., Nigay, L.: Visual menu techniques. *ACM Comput. Surv. (CSUR)* **49**(4), 1–41 (2016)
- Belli, L., Cirani, S., Gorrieri, A., Picone, M.: A novel smart object-driven ui generation approach for mobile devices in the internet of things. In: Proceedings of the 1st international workshop on experiences with the design and implementation of smart objects, pp. 1–6 (2015)
- Berman, A., Thakare, K., Howell, J., Quek, F., Kim, J.: Howdiy: towards meta-design tools to support anyone to 3d print anywhere. In: 26th International Conference on Intelligent User Interfaces, pp. 491–503 (2021)
- Bosch. Bosch smart home (2022)
- Brdnik, S., Heričko, T., Šumak, B.: Intelligent user interfaces and their evaluation: a systematic mapping study. *Sensors* **22**(15), 5830 (2022)
- Campos, P.G., Bellogín, A., Díez, F., Chavarriaga, J.E.: Simple time-biased knn-based recommendations. In: Proceedings of the Workshop on Context-Aware Movie Recommendation, pp. 20–23 (2010)
- Carrera-Rivera, A., Larrinaga, F., Lasa, G.: Context-awareness for the design of smart-product service systems: Literature review. *Comput. Ind.* **142**, 103730 (2022)
- Carrera-Rivera, A., Reguera-Bakhache, D., Larrinaga, F., Lasa, G., Garitano, I.: Structured dataset of human-machine interactions enabling adaptive user interfaces. *Scientific Data* **10**(1), 831 (2023)
- Champiri, Z.D., Mujtaba, G., Salim, S.S., Chong, C.Y.: User experience and recommender systems. In: 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1–5. IEEE (2019)
- Chiu, M.C., Huang, J.H., Gupta, S., Akman, G.: Developing a personalized recommendation system in a smart product service system based on unsupervised learning model. *Comput. Ind.* **128**, 103421 (2021)
- Chou, J.R.: A TRIZ-based product-service design approach for developing innovative products. *Comput. Ind. Eng.* **161**, 107608 (2021)
- Cong, J., Zheng, P., Bian, Y., Chen, C.H., Li, J., Li, X.: A machine learning-based iterative design approach to automate user satisfaction degree prediction in smart product-service system. *Comput. Ind. Eng.* **165**, 107939 (2022). <https://doi.org/10.1016/j.cie.2022.107939>
- Cong, J.C., Chen, C.H., Zheng, P., Li, X., Wang, Z.: A holistic relook at engineering design methodologies for smart product-service systems development. *J. Clean. Prod.* **272**, 122737 (2020). <https://doi.org/10.1016/j.jclepro.2020.122737>
- De Nicola, A., Missikoff, M., Navigli, R.: A proposal for a unified process for ontology building: Upon. In: Database and Expert Systems Applications: 16th International Conference, DEXA 2005, Copenhagen, Denmark, August 22-26, 2005. Proceedings 16, pp. 655–664. Springer (2005)
- Dong, Y., Zhu, R., Tian, Q., Liu, W., Peng, W.: A Scenario Interaction-centered Conceptual Information Model for UX Design of User-oriented Product-service System. *Procedia CIRP* **83**, 335–338 (2019)
- Esheiba, L., Elgammal, A., Helal, I.M., El-Sharkawi, M.E.: A hybrid knowledge-based recommender for product-service systems mass customization. *Information* **12**(8), 296 (2021)
- Fernandez-Garcia, A.J., Iribarne, L., Corral, A., Wang, J.Z.: Evolving mashup interfaces using a distributed machine learning and model transformation methodology. In: On the Move to Meaningful Internet Systems: OTM 2015 Workshops: Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, E12N, FBM, INBAST, ISDE, META4eS, and MSC 2015, Rhodes, Greece, October 26-30, 2015. Proceedings, pp. 401–410. Springer (2015)
- Fitbit. Fitbit premium services (2023)
- Gan, R., Liang, J., Ahmad, B.I., Godsill, S.: Modeling intent and destination prediction within a bayesian framework: predictive touch as a usecase. *Data-Centric Eng.* **1**, e12 (2020)

- Gil, Y., Garijo, D., Khider, D., Knoblock, C.A., Ratnakar, V., Osorio, M., Vargas, H., Pham, M., Pujara, J., Shbita, B., et al.: Artificial intelligence for modeling complex systems: taming the complexity of expert models to improve decision making. *ACM Trans. Interact. Intell. Syst.* **11**(2), 1–49 (2021)
- Gobert, C., Todi, K., Bailly, G., Oulasvirta, A.: Sam: a modular framework for self-adapting web menus. In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 481–484 (2019)
- Gonçalves, T.G., da Rocha, A.R.C.: Development process for intelligent user interfaces: an initial approach. In: *Proceedings of the XVIII Brazilian Symposium on Software Quality*, pp. 210–215 (2019)
- Gong, C.: Human-machine interface: Design principles of visual information in human-machine interface design. In: *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, Volume 2, pp. 262–265. IEEE (2009)
- Guarino, N., Oberle, D., Staab, S.: What is an ontology?, *Handbook on ontologies*, 1–17. Springer (2009)
- Haugeland, I.K.F., Følstad, A., Taylor, C., Bjørkli, C.A.: Understanding the user experience of customer service chatbots: an experimental study of chatbot interaction design. *Int. J. Hum Comput Stud.* **161**, 102788 (2022)
- Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inform. Syst. (TOIS)* **22**(1), 5–53 (2004)
- Hussain, J., Ul Hassan, A., Muhammad Bilal, H.S., Ali, R., Afzal, M., Hussain, S., Bang, J., Banos, O., Lee, S.: Model-based adaptive user interface based on context and user experience evaluation. *J. Multimodal User Interfaces* **12**, 1–16 (2018)
- Iqbal, M.W., Ahmad, N., Shahzad, S.K., Feroz, I., Mian, N.A.: Towards adaptive user interfaces for mobile-phone in smart world. *International journal of advanced computer science and applications* **9**(11) (2018)
- Iqbal, M.W., Ch, N.A., Shahzad, S.K., Naqvi, M.R., Khan, B.A., Ali, Z.: User context ontology for adaptive mobile-phone interfaces. *IEEE Access* **9**, 96751–96762 (2021)
- Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: *Recommender systems: an introduction*. Cambridge University Press (2010)
- Janowicz, K., Haller, A., Cox, S.J., Le Phuoc, D., Lefrançois, M.: Sosa: a lightweight ontology for sensors, observations, samples, and actuators. *J. Web Semant.* **56**, 1–10 (2019)
- Kaminskas, M., Bridge, D.: Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst. (TiS)* **7**(1), 1–42 (2016)
- Khan, I., Khusro, S.: Towards the design of context-aware adaptive user interfaces to minimize drivers' distractions. *Mob. Inf. Syst.* (2020). <https://doi.org/10.1155/2020/8858886>
- Kuhlenkötter, B., Wilkens, U., Bender, B., Abramovici, M., Süße, T., Göbel, J., Herzog, M., Hypki, A., Lenkenhoff, K.: New perspectives for generating smart PSS solutions-life cycle, methodologies and transformation. *Procedia CIRP* **64**, 217–222 (2017)
- Kulkarni, S., Rodd, S.F.: Context aware recommendation systems: a review of the state of the art techniques. *Comput. Sci. Rev.* **37**, 100255 (2020)
- Landowska, A., Szwoch, M., Szwoch, W.: Methodology of affective intervention design for intelligent systems. *Interact. Comput.* **28**(6), 737–759 (2016)
- Le, D.T., Thi, T.T.P., Pham-Nguyen, C., Nam, L.N.H.: Towards a context-aware knowledge model for smart service systems. In: *International Conference on Computational Collective Intelligence*, pp. 767–778. Springer (2020)
- Lenze. *Lenze smart apps* (2022)
- Li, G., Wang, L., Ou, W.: Robust personalized ranking from implicit feedback. *Int. J. Pattern Recognit Artif Intell.* **30**(01), 1659001 (2016)
- Li, X., Chen, C.H., Zheng, P., Wang, Z., Jiang, Z., Jiang, Z.: A knowledge graph-aided concept-knowledge approach for evolutionary smart product-service system development. *J. Mech. Des.* **142**(10), 101403 (2020)
- Lin, K.Y., Chien, C.F., Kerh, R.: Unison framework of data-driven innovation for extracting user experience of product design of wearable devices. *Comput. Ind. Eng.* **99**, 487–502 (2016)
- Liu, N.N., Xiang, E.W., Zhao, M., Yang, Q.: Unifying explicit and implicit feedback for collaborative filtering. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1445–1448 (2010)
- Liu, W., Li, X., Huang, D.: A survey on context awareness. In: *2011 International Conference on Computer Science and Service System (CSSS)*, pp. 144–147. IEEE (2011)



- Liu, Z., Ming, X., Song, W., Qiu, S., Qu, Y.: A perspective on value co-creation-oriented framework for smart product-service system. *Procedia Cirp* **73**, 155–160 (2018)
- Machado, E., Singh, D., Cruciani, F., Chen, L., Hanke, S., Salvago, F., Kropf, J., Holzinger, A.: A conceptual framework for adaptive user interfaces for older adults. In: 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 782–787. IEEE (2018)
- Maleki, E., Belkadi, F., Boli, N., Van Der Zwaag, B.J., Alexopoulos, K., Koukas, S., Marin-Perianu, M., Bernard, A., Mourtzis, D.: Ontology-based framework enabling smart product-service systems: application of sensing systems for machine health monitoring. *IEEE Internet Things J.* **5**(6), 4496–4505 (2018)
- McComb, C., Cagan, J., Kotovsky, K.: Capturing human sequence-learning abilities in configuration design tasks through markov chains. *J. Mech. Des.* **139**(9), 091101 (2017)
- Mezhoudi, N., Khaddam, I., Vanderdonck, J.: Toward usable intelligent user interface. In: Human-Computer Interaction: Interaction Technologies: 17th International Conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part II 17, pp. 459–471. Springer (2015)
- Miraz, M.H., Ali, M., Excell, P.S.: Adaptive user interfaces and universal usability through plasticity of user interface design. *Comput. Sci. Rev.* **40**, 100363 (2021)
- Mitchell, J., Shneiderman, B.: Dynamic versus static menus: an exploratory comparison. *ACM SigCHI Bulletin* **20**(4), 33–37 (1989)
- Mourtzis, D., Fotia, S., Vlachou, E., Koutoupes, A.: A Lean PSS design and evaluation framework supported by KPI monitoring and context sensitivity tools. *Int. J. Adv. Manuf. Technol.* **94**(5–8), 1623–1637 (2018)
- Nguyen, L., et al.: A new aware-context collaborative filtering approach by applying multivariate logistic regression model into general user pattern. *J. Data Anal. Inform. Proc.* **4**(03), 124 (2016)
- Ochoa, W., Larrinaga, F., Pérez, A.: Context-aware workflow management for smart manufacturing: a literature review of semantic web-based approaches. *Futur. Gener. Comput. Syst.* (2023). <https://doi.org/10.1016/j.future.2023.03.017>
- Oestreich, H., Heinz-Jakobs, M., Sehr, P., Wrede, S.: Human-centered adaptive assistance systems for the shop floor, *Human-Technology Interaction: Shaping the Future of Industrial User Interfaces*, 83–125. Springer (2022)
- Orghidan, R., Gordan, M., Danciu, M., Vlaicu, A.: A prototype for the creation and interactive visualization of 3d human face models. In: *Advanced Engineering Forum*, Volume 8, pp. 45–54. Trans Tech Publ (2013)
- Ozok, A.A., Fan, Q., Norcio, A.F.: Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: results from a college student population. *Behav. Inf. Technol.* **29**(1), 57–83 (2010)
- Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., Fragopoulou, P.: Collaborative filtering recommender systems taxonomy. *Knowl. Inf. Syst.* **64**(1), 35–74 (2022)
- Peck, E.M., Easse, E., Marshall, N., Stratton, W., Perrone, L.F.: Flyloop: a micro framework for rapid development of physiological computing systems. In: *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pp. 152–157 (2015)
- Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: a survey. *IEEE Commun. Surv. Tutor.* **16**(1), 414–454 (2013)
- Pezzotta, G., Cavalieri, S., Romero, D.: Collaborative product-service systems engineering: Towards an active role of customers and stakeholders in value co-creation. In: *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pp. 1247–1254. IEEE (2017)
- Pirola, F., Boucher, X., Wiesner, S., Pezzotta, G.: Digital technologies in product-service systems: a literature review and a research agenda. *Comput. Ind.* **123**, 103301 (2020). <https://doi.org/10.1016/j.compind.2020.103301>
- Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: *Proceedings of the fifth ACM conference on Recommender systems*, pp. 157–164 (2011)
- Pulikottil, T., Estrada-Jimenez, L.A., Abadía, J.J.P., Carrera-Rivera, A., Torayev, A., Rehman, H.U., Mo, F., Nikghadam-Hojjati, S., Barata, J.: Big data life cycle in shop-floor-trends and challenges. *IEEE Access* (2023). <https://doi.org/10.1109/ACCESS.2023.3253286>
- Raza, S., Ding, C.: Progress in context-aware recommender systems-an overview. *Comput. Sci. Rev.* **31**, 84–97 (2019)

- Reguera-Bakhache, D., Garitano, I., Uribeetxeberria, R., Cernuda, C., Zurutuza, U.: Data-driven industrial human-machine interface temporal adaptation for process optimization. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Volume 1, pp. 518–525. IEEE (2020)
- Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., Van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25–29, 2014. Proceedings 11, pp. 752–767. Springer (2014)
- Ricci, F., Rokach, L., Shapira, B.: *Recommender Systems Handbook*, pp. 1–35. Boston, MA: Springer US (2011)
- Rodrigues, J.M., Pereira, J.A., Sardo, J.D., de Freitas, M.A., Cardoso, P.J., Gomes, M., Bica, P.: Adaptive card design ui implementation for an augmented reality museum application. In: Universal Access in Human-Computer Interaction. Design and Development Approaches and Methods: 11th International Conference, UAHCI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017, Proceedings, Part I 11, pp. 433–443. Springer (2017)
- Seo, D.W., Kim, H., Kim, J.S., Lee, J.Y.: Hybrid reality-based user experience and evaluation of a context-aware smart home. *Comput. Ind.* **76**, 11–23 (2016)
- Smirnov, A.V., Kashevnik, A.M., Ponomarev, A.: Context-based infomobility system for cultural heritage recommendation: tourist assistant-tais. *Pers. Ubiquit. Comput.* **21**, 297–311 (2017)
- Stefanidi, Z., Margetis, G., Ntoa, S., Papagiannakis, G.: Real-time adaptation of context-aware intelligent user interfaces, for enhanced situational awareness. *IEEE Access* **10**, 23367–23393 (2022)
- Stephanidis, C.: Adaptive techniques for universal access. *User Model. User-Adap. Inter.* **11**, 159–179 (2001)
- Stephanidis, C., Paramythis, A., Sfyarakis, M., Stergiou, A., Maou, N., Leventis, A., Paparoulis, G., Karagianidis, C.: Adaptable and adaptive user interfaces for disabled users in the avanti project. In: Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services: 5th International Conference on Intelligence in Services and Networks, IS&N'98 Antwerp, Belgium, May 25–28, 1998 Proceedings 5, pp. 153–166. Springer (1998)
- Stumpf, S.: Horses for courses: Making the case for persuasive engagement in smart systems. In: Joint Proceedings of the ACM IUI 2019 Workshops, Volume 2327. CEUR (2019)
- Tahir, R.: Analyzing the intelligence in user interfaces. In: 2015 SAI Intelligent Systems Conference (IntelliSys), pp. 674–680. IEEE (2015)
- Tan, H.Z., Zhao, W., Shen, H.H.: Adaptive user interface optimization for multi-screen based on machine learning. In: 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)), pp. 743–748. IEEE (2018)
- Tarus, J.K., Niu, Z., Yousif, A.: A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Futur. Gener. Comput. Syst.* **72**, 37–48 (2017)
- ThoughtWorks. Thoughtworks technology radar, vol. 27. (2022) [https://www.thoughtworks.com/content/dam/thoughtworks/documents/radar/2022/10/tr\\_technology\\_radar\\_vol\\_27\\_en.pdf](https://www.thoughtworks.com/content/dam/thoughtworks/documents/radar/2022/10/tr_technology_radar_vol_27_en.pdf). [Online; accessed 19-June-2023]
- Tsandilas, T., Schraefel, M.: An empirical assessment of adaptation techniques. In: CHI'05 Extended Abstracts on Human Factors in Computing Systems, pp. 2009–2012 (2005)
- Valencia Cardona, A., Mugge, R., Schoormans, J.P., Schifferstein, H.N.: Challenges in the design of smart product-service systems (PSSs): Experiences from practitioners. In: Proceedings of the 19th DMI: Academic Design Management Conference. Design Management in an Era of Disruption, London, UK, September 2–4, 2014. Design Management Institute (2014)
- van Ooijen, I., Vrabec, H.U.: Does the GDPR enhance consumers' control over personal data? An analysis from a behavioural perspective. *J. Consum. Policy* **42**(1), 91–107 (2019)
- Völkel, S.T., Schneegass, C., Eiband, M., Buschek, D.: What is "intelligent" in intelligent user interfaces? a meta-analysis of 25 years of iui. In: Proceedings of the 25th international conference on intelligent user interfaces, pp. 477–487 (2020)
- Wallace, J.: A holistic ux methodological framework for measuring the aspects of how dynamic, adaptive and intelligent a software solution is and make recommendations for improvement (2020)
- Wang, Z., Chen, C.H., Li, X., Zheng, P., Khoo, L.P.: A context-aware concept evaluation approach based on user experiences for smart product-service systems design iteration. *Adv. Eng. Inform.* **50**, 101394 (2021)

- Wang, Z., Chen, C.H., Zheng, P., Li, X., Khoo, L.P.: A novel data-driven graph-based requirement elicitation framework in the smart product-service system context. *Adv. Eng. Inform.* **42**, 100983 (2019)
- Wang, Z., Chen, C.H., Zheng, P., Li, X., Khoo, L.P.: A graph-based context-aware requirement elicitation approach in smart product-service systems. *Int. J. Prod. Res.* **59**(2), 635–651 (2021)
- Wang, Z., Chen, C.H., Zheng, P., Li, X., Song, W.: A hypergraph-based approach for context-aware smart product-service system configuration. *Comput. Ind. Eng.* **163**, 107816 (2022)
- Wang, Z., Li, X., Chen, C.H., Khoo, L.P.: Evaluating Smart PSS Solutions with Context-Awareness in Usage Phase. *Transdisciplinary Engineering for Complex Socio-technical Systems - Real-life Applications*, Volume 12 of *Advances in Transdisciplinary Engineering*, 333–342. IOS Press. ISSN: 2352751X, 23527528 (2020, September). <https://doi.org/10.3233/ATDE200092>
- Wang, Z., Zheng, P., Chen, C.H., Khoo, L.P.: A survey of requirements management in smart product-service systems, *Transdisciplinary Engineering for Complex Socio-technical Systems*, 613–622. IOS Press (2019)
- Wattarachchi, W.D., Hewagamage, K., Hettiarachchi, E.: A framework to decide adaptive functionalities by considering user emotions and the context. In: 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 178–183. IEEE (2020)
- Wenjuan, H., Premalatha, R., Aiswarya, R.: Physical education system and training framework based on human–computer interaction for augmentative and alternative communication. *International Journal of Speech Technology*: 1–11 (2022)
- Wetter-Edman, K., Sangiorgi, D., Edvardsson, B., Holmlid, S., Grönroos, C., Mattelmäki, T.: Design for value co-creation: exploring synergies between design for service and service logic. *Serv. Sci.* **6**(2), 106–121 (2014)
- Yigitbas, E., Jovanovikj, I., Biermeier, K., Sauer, S., Engels, G.: Integrated model-driven development of self-adaptive user interfaces. *Softw. Syst. Model.* **19**, 1057–1081 (2020)
- Yigitbas, E., Sauer, S., Engels, G.: Adapt-ui: an ide supporting model-driven development of self-adaptive uis. In: Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems, pp. 99–104 (2017)
- Yu, Y., Sung, T.J.: A value-based view of the smart pss adoption: a study of smart kitchen appliances. *Serv. Bus.* (2023). <https://doi.org/10.1007/s11628-023-00529-9>
- Yuan, W., Chang, D., Han, T.: A context-aware smart product-service system development approach and application case. *Comput. Ind. Eng.* **183**, 109468 (2023)
- Zhang, X., Chen, L., Sheng, G., Lu, X., Ming, X.: An innovation service system and personalized recommendation for customer-product interaction life cycle in smart product service system. *J. Clean. Prod.* **398**, 136470 (2023)
- Zheng, P., Wang, Z., Chen, C.H., Khoo, L.P.: A survey of smart product-service systems: key aspects, challenges and future perspectives. *Adv. Eng. Inform.* **42**, 100973 (2019)
- Zhou, J., Sun, J., Chen, F., Wang, Y., Taib, R., Khawaji, A., Li, Z.: Measurable decision making with gsr and pupillary analysis for intelligent user interface. *ACM Trans. Comput. Human Interact. (ToCHI)* **21**(6), 1–23 (2015)
- Zhou, X., Peng, X., Xie, T., Sun, J., Ji, C., Liu, D., Xiang, Q., He, C.: Latent error prediction and fault localization for microservice applications by learning from system trace logs. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 683–694 (2019)
- Ziarani, R.J., Ravanmehr, R.: Serendipity in recommender systems: a systematic literature review. *J. Comput. Sci. Technol.* **36**, 375–396 (2021)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Angela Carrera-Rivera** received her B.Sc. degree in Information Systems from Escuela Superior Politécnica del Litoral, Ecuador, and her M.Sc. degree in Information Technology Engineering from the University of Melbourne, Australia. She holds a Ph.D. degree in Applied Engineering from Mondragon University, where her research focused on data-driven user interface adaptations. Her research interests include user experience, data analysis, machine learning, and data science. Angela has solid experience in data analysis and data science, with a proven track record in designing and implementing data-based solutions.

**Dr. Felix Larrinaga** was born in Markina-Xemein (Bizkaia) Spain in 1965. He obtained his Ph.D. in Mobile Communications at Staffordshire University (UK) in 1999. Previously, he finished the B.Sc. (Hons) in Industrial Electronics at Mondragon University in 1995. Since 2000, he combines lecturing and research at Mondragon University with an extensive experience in industrial work as a project leader. He is currently a lecturer and a researcher at Mondragon University (Basque Country, Spain). He has actively participated in the creation of the new research team on Web Engineering at the Faculty of Engineering. The research activities of the group are in the areas of interoperability, web architecture, microservices, and semantic web technologies. The team works on multidisciplinary projects with other universities, technological centers, and enterprises in areas such as energy, health, or industry. At the moment, Dr. Larrinaga is working on several projects related to manufacturing, smart cities, and energy.

**Dr. Ganix Lasa** is a lecturer and researcher in Innovation in Industrial Design at the Engineering School of Mondragon Unibertsitatea. He holds a PhD degree in User Experience Evaluation from Mondragon Unibertsitatea. His research focuses on interaction design, user experience and product development with a User-Centred Design approach. He is the author of several scientific articles referring to technology of interaction design, user experience and product development.

**Giovanna Martinez-Arellano** is a Senior Research Fellow at the Institute for Advanced Manufacturing, University of Nottingham, UK. With a background in Computer Science, Giovanna focuses on the application of machine learning approaches for digital and smart manufacturing. She is the recipient of the Anne McLaren Research Fellowship Award, which focuses on the development of scalable and robust machine learning models for complex reconfigurable systems.

**Gorka Unamuno** is a research engineer skilled in Robotics and Automation systems with more than 20 years of experience. Expertise in the application of informatics to control automation. Currently, he is developing his career at ICT and automation research group within IDEKO, contributing thus to the development of next generation industrial production systems. He is graduated at Mondragon University (Spain) with a bachelor's degree in Electronical engineering in year 2003 and a masters degree in 2020 in the university of Basque country. He has participated actively in many research projects funded with public programmes (EC, national, regional). Proven experience in industry-driven projects, including both research work as well as industrial implementation of advanced production systems with a high technological background.