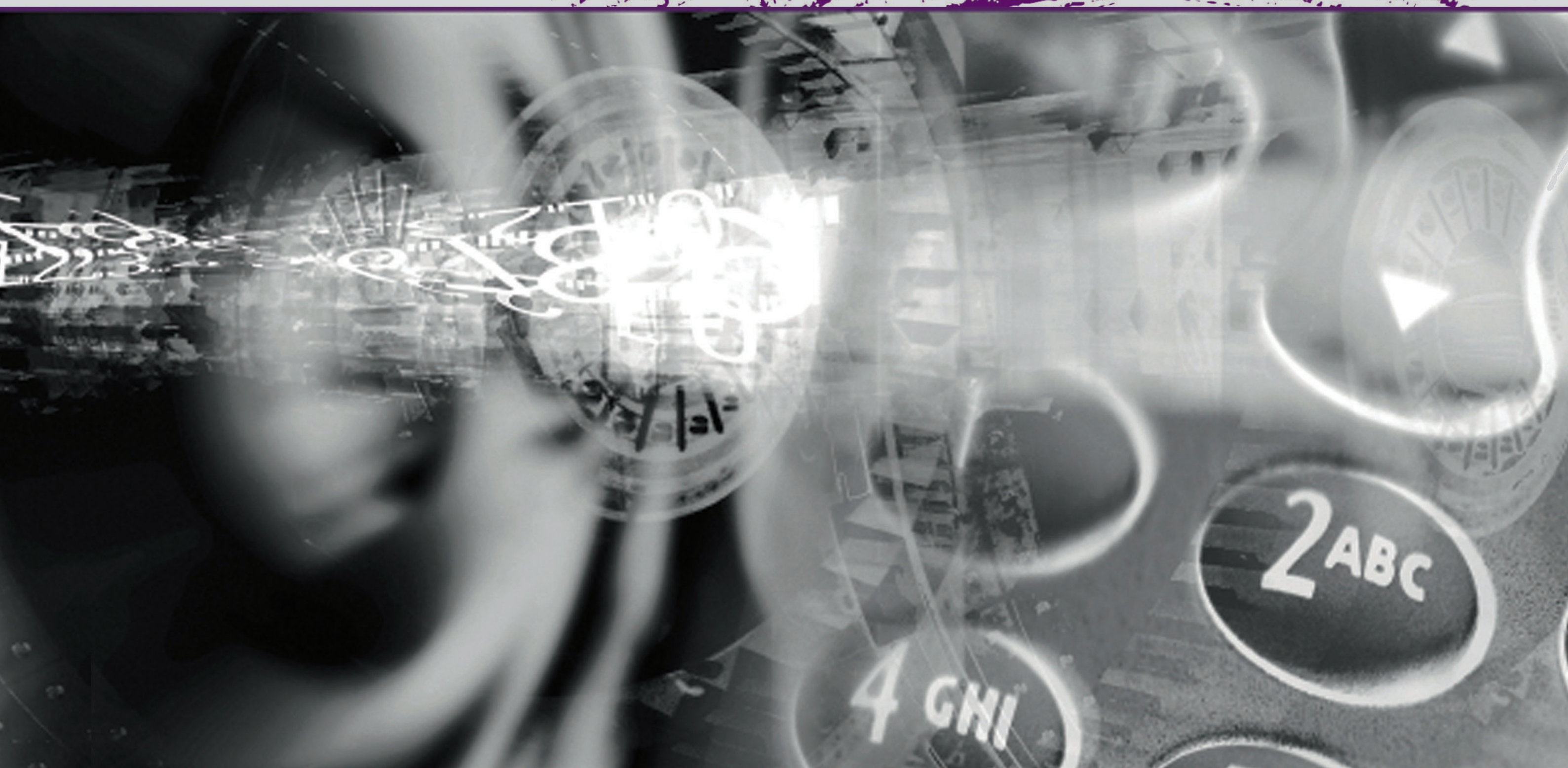
Mondragon
Unibertsitatea

DOCTORAL THESIS

ADAPTUI: A CONTEXT-AWARE FRAMEWORK FOR ADAPTIVE USER INTERFACES IN SMART-PRODUCT SERVICE SYSTEMS (S-PSS)



ANGELA ISABEL CARRERA RIVERA | Arrasate-Mondragón, 2024

Mondragon Unibertsitatea Goi Eskola Politeknikoa

AdaptUI : A context-aware framework for adaptive user interfaces in Smart Product-Service Systems (S-PSS)

Angela Carrera Rivera

2024

Doctoral Program in Applied Engineering

Title of the doctoral thesis:

AdaptUI : A context-aware framework for adaptive user interfaces in Smart Product-Service Systems (S-PSS)

Presented by

Angela Carrera Rivera

Electronics and Computer Science department

Mondragon Goi Eskola Politeknikoa

Supervised by

Dr. Felix Larrinaga

Dr. Ganix Lasa

In Arrasate, 2024

Cobb: You create the world of the dream, you bring the subject into that dream, and they fill it with their subconscious.

Ariadne: How could I ever acquire enough detail to make them think that its reality?

Cobb: Well dreams, they feel real while we're in them, right? It's only when we wake up that we realize how things are actually strange. Let me ask you a question, you, you never really remember the beginning of a dream do you? You always wind up right in the middle of what's going on.

Ariadne: I guess, yeah.

Cobb: So how did we end up here?

Ariadne: Well we just came from the a...

Cobb: Think about it Ariadne, how did you get here? Where are you right now?

Ariadne: We're dreaming?

Cobb: You're actually in the middle of the workshop right now, sleeping. This is your first lesson in shared dreaming. Stay calm.

— Inception, 2010

Acknowledgments

"Changing is living." There is no life without change, whether small or large; we undergo transformations daily. Although I tend to resist change and stepping out of my comfort zone, I am immensely grateful for this experience. I express my profound gratitude for the opportunity to be part of the DimanD project, to which I extend my congratulations to the team, especially to Miren, Felix, and Leire, who have worked really hard to provide us with this experience. This journey has given me the chance to learn and grow, as well as to get to know colleagues with whom I have not only shared in the professional realm but also had the privilege to connect with personally – Miriam, Joaquin, Nathaly, Luis, Agajan, Fabio, Jose, Trunal, Terrin, Fan, Hammood. I hope we stay in touch!

Igualmente, agradezco inmensamente a mis supervisores Félix y Ganix, sin quienes no estaría escribiendo este documento. Gracias por no solo guiarme en mi trabajo, sino también por asegurarse de que me sintiera bien acogida. También quiero expresar mi gratitud a todos los que me han ayudado de una u otra manera en esta tesis. Agradezco sinceramente a IDEKO y a todo el equipo, en particular a Gorka, Ander y Jose Luis, por brindarme la oportunidad de llevar a cabo una estancia fundamental para este proyecto y por su valiosa ayuda. De igual manera, agradezco a la Universidad de Nottingham y a su equipo por permitirme realizar mi estancia en este centro y por hacerme sentir bien recibida. Especial agradecimiento a Giovanna, quien estuvo siempre pendiente y me proporcionó excelentes comentarios que contribuyeron a mejorar mi trabajo.

Deseo expresar tambien mi agradecimiento infinito a mi familia: a mis padres, Abdon y Ángela, y a mis hermanos, Gabriel y Andrés. A pesar de encontrarse a "solo" 8,000 kilómetros de distancia en Ecuador, siempre han estado ahí para mí. A mis amigos Jennifer y David, les agradezco sus mensajes, su capacidad para sacarme sonrisas y por nunca dejarme fuera de su día a día, a pesar de la distancia. Finalmente, quiero dirigir unas palabras a William, mi esposo, quien siempre ha creído en mí incluso más de lo que yo misma lo hago. No estaría en este punto si no fuera por tu constante apoyo y por compartir este sueño conmigo, del cual llego la hora de despertar :)

Statement of Originality

Hereby I, Angela Carrera Rivera declare that this thesis is the result of my personal work, and that it has not been previously submitted to obtain another degree or professional qualification. The ideas, formulations, images, illustrations taken from other sources have been duly cited and referenced.

Yo Angela Carrera Rivera declaro que el trabajo de investigación desarrollado en esta tesis es original, fruto de mi trabajo personal, y que no ha sido previamente presentado para obtener otro título o calificación profesional. Las ideas, formulaciones, imágenes, ilustraciones tomadas de fuentes ajenas han sido debidamente citadas y referenciadas.

Arrasate, 2024

Angela Carrera Rivera

Abstract

Smart Product-Service Systems (S-PSS) embody a transformative business model that integrates intelligent products with advanced digital capabilities and associated e-services. This dynamic model holds potential for driving sustainable practices. Despite this potential, fully exploiting digital capabilities and optimizing user experiences within S-PSS remains a challenge.

In order to address the limitations of conventional User Experience (UX) analysis, which often concentrates on design stages and overlooks evolving user needs during usage, this thesis proposes a practical framework for developing adaptive user interfaces within S-PSS. The framework seamlessly integrates ontologies and context-aware recommendation systems, utilizing user interactions as the primary data source. This integration facilitates the development of adaptive user interfaces, ensuring digital services dynamically respond to user preferences and contextual cues.

A key contribution of this work lies in the comprehensive integration of various components, resulting in the creation of adaptive user interfaces tailored for digital services. The practical application of the framework is illustrated through two case studies involving a smart device app and an industrial scenario. This real-world exploration serves to demonstrate the effective implementation of the proposed framework in diverse contexts, as the hands-on development approach considers technological aspects and utilizes appropriate tools, providing valuable evidence into the practicality and effectiveness of the framework.

The evaluation first focuses on assessing the recommendation engine based on collaborative filtering. These results highlight the enhanced precision of recommendations when employing a context-aware approach compared to a traditional approach. Subsequently, pragmatic aspects of UX, such as usefulness and system efficiency, are evaluated based on participant feedback and UX metrics. The findings reveal an overall positive impact on the utilization of the smart product-services.

Laburpena

Smart Product-Service Systems (S-PSS) produktu adimendunak gaitasun digital aurreratuekin eta lotutako zerbitzuekin integratzen dituen negozio-eredu eraldatzailea adierazten du, iraunkortasunerako praktikak gidatzen dituena. Potentzial hori izan arren, gaitasun digitalak guztiz aprobetxatzea eta S-PSS-en erabiltzaileen esperientziak optimizatzea erronka izaten jarraitzen du.

Erabiltzaile-esperientziaren (UX) ohiko analisiaren mugak gainditzeko, askotan diseinu-faseetan zentratzen dena eta erabileran zehar erabiltzailearen beharrak aldakorrak alde batera uzten dituena, tesi honek S-PSS barruan erabiltzaile-interfaze moldagarriak garatzeko esparru praktiko bat aurkezten du. Esparru praktikoak ondo integratzen ditu ontologiak eta testuinguruari buruzko gomendio sistemak, erabiltzaileen interakzioak erabiliz datu-iturri nagusi gisa. Integrazio honek erabiltzaile-interfaze moldagarriak garatzea errazten du, zerbitzu digitalek erabiltzailearen hobespenei eta testuinguruko seinaleei modu dinamikoan erantzuten dietela bermatuz.

Lan honen funtsezko ekarpena hainbat osagairen integrazioan datza, eta horren ondorioz zerbitzu digitaletarako diseinatutako erabiltzaile-interfaze moldagarriak sortu dira. Markoaren aplikazio praktikoa gailu adimendunaren aplikazioa eta industria-eszenatoki bat duten bi kasu-azterketen bidez azaltzen da. Mundu errealeko esplorazio honek proposatutako esparruaren ezarpen eraginkorra hainbat testuingurutan frogatzeko balio du, ezarpenak eta garapenak alderdi teknologikoak kontuan hartzen baititu, tresna egokiak erabiliz, esparruaren bideragarritasunari eta eraginkortasunari buruzko frogak emanez.

Ebaluazio-emaitzak lehenik eta behin "kolaborazio-iragazkia"n oinarritutako gomendio-motorean oinarritzen dira, gomendioen zehaztasun handiagoa nabarmenduz, "testuinguruaren jabe den" ikuspegia erabiliz. Ondoren, UXren alderdi pragmatikoak, hala nola sistemaren erabilgarritasuna eta eraginkortasuna, parte-hartzaileen feedbackaren eta UX neurketen bidez ebaluatzen dira, gailu adimendunaren erabileran eragin positiboa dela agerian utziz.

Resumen

Los Sistemas de Producto-Servicio Inteligentes (S-PSS) representan un modelo de negocio transformador que integra productos inteligentes con capacidades digitales avanzadas y servicios asociados, potencialmente orientando las prácticas hacia la sostenibilidad. A pesar de este potencial, aprovechar completamente las capacidades digitales y optimizar las experiencias de usuario dentro de los S-PSS sigue siendo un desafío.

Para superar las limitaciones del análisis convencional de Experiencia de Usuario (UX), que a menudo se centra en las etapas de diseño y pasa por alto las necesidades cambiantes del usuario durante el uso, en esta tesis se presenta un marco práctico para el desarrollo de interfaces de usuario adaptativas dentro de los S-PSS. El marco practico integra de manera fluida ontologías y sistemas de recomendación conscientes del contexto, utilizando las interacciones del usuario como la principal fuente de datos. Esta integración facilita el desarrollo de interfaces de usuario adaptativas, asegurando que los servicios digitales respondan dinámicamente a las preferencias del usuario y las señales contextuales.

Una contribución clave de este trabajo radica en la integración de varios componentes, lo que resulta en la creación de interfaces de usuario adaptativas diseñadas para servicios digitales. La aplicación práctica del marco se ilustra a través de dos casos de estudio que involucran una aplicación para dispositivos inteligentes y un escenario industrial. Esta exploración del mundo real sirve para demostrar la implementación efectiva del marco propuesto en contextos diversos, ya que la implementación y desarrollo considera aspectos tecnológicos, utilizando herramientas apropiadas, proporcionando evidencia sobre la viabilidad y eficacia del marco de trabajo.

Los resultados de la evaluación se enfocan primero en el motor de recomendación basado en "filtro colaborativo", destacando la mayor precisión de las recomendaciones al emplear un enfoque "consciente del contexto". Posteriormente, se evalúan aspectos pragmáticos de la UX, como la utilidad y eficiencia del sistema, a través de la retroalimentación de los participantes y métricas de UX, revelando un impacto general positivo en el uso del dispositivo inteligente.

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. xiv, 23, 26, 56

AR Augmented Reality. xiv, 40

AUI Adaptive User Interface. vii, xi, xiii, xiv, 5, 6, 10, 11, 23–25, 51, 56, 59, 68, 70, 71, 77, 80, 82–84, 86, 89, 93, 108, 110, 115–120, 125, 126, 131, 133, 134

AUIs Adaptive User Interfaces. xiv, 5, 7, 10, 23, 26, 56, 99, 120, 125

CARS Context-Aware Recommendation Systems. xiii, xiv, 25, 45–47

CF Collaborative Filtering. xiv, 66

CPS Cyber-Physical System. xiv, 12, 16, 21, 22, 26

CTR Click Through Rate. xiv, 76, 88, 95, 97, 98

DimanD Digital Manufacturing and Design. xiv, 2, 133

ECG Electrocardiogram. xiv, 39, 40

EEG Electroencephalography. xiv, 39, 40

fNIRS Functional Near Infrared Spectroscopy. xiv, 39, 40

GA4 Google Analytics 4. xiv, 86

HCI Human-Computer Interfaces. xiv

HTML Hypertext Markup Language. xiv

ICT Information and Communications Technology. xiv, 1

IoT Internet of things. xiv, 84

KNN K-Nearest Neighbors. xiv, 66

MAE Mean Average Error. xiv, 74, 83, 90, 92, 113, 114, 130

MRR Mean Reciprocal Rank. xiv, 74

NLP Natural Language Processing. xiv, 41, 45, 50

OOP Object Oriented Programming. xiv, 20

OWL Web Ontology Language. xiv, 19

PSS Product-Service Systems. xiv, 1, 11, 12, 14, 26

RS Recommendation Systems. xi, xiv, 25, 45, 47, 65, 87, 97

S-D Service-Dominant. xiv, 36

S-PSS Smart Product-Service Systems. xiii, xiv, 1–6, 8, 10–15, 26–30, 32–34, 47, 68, 84, 125, 132

SCP Smart Connected Products. xiv, 42, 58, 60, 84

SCPs Smart Connected Products. xiv, 57

SLR Systematic Literature Review. x, xiv, 6, 26–29, 31, 36, 45, 52, 55, 125

UCD User-centred design. xiv, 3, 27, 29, 32, 47, 50, 52, 53, 56, 80

UI User Interface. xiv, 4, 23, 47, 58, 61, 68–71, 78, 84–86, 88, 89, 100, 115, 118, 123, 128

UID User Interface Descriptor. xi, xiv, 70, 71, 79, 112

UIs User Interfaces. xiv, 3

UX User experience. xiv, 2, 3, 5–7, 10, 26–30, 38, 47, 52, 76, 90, 100, 102, 111, 127

VR Virtual Reality. xiv, 40

1
Introduction

The impact of digital technologies on everyday life has been profound. This influence extends into the industrial domain, where Industry 4.0 (i4.0) is an evidence of the transformative power of Information and Communications Technology (ICT). Initially, ICT optimized internal and organizational processes, enhancing traditional product and service offerings. However, the evolution of business models, the arrival of new technologies, and shifting consumer expectations have driven ICT beyond organizational boundaries, and move companies toward servitization.

Servitization is a process that has reshaped manufacturing companies and involves the strategic integration of products with related services to meet user needs (Tukker, 2004). This evolution has resulted on the origin of Product-Service Systems (PSS), where the bundling of products and services maximizes both product value and business profits (Hallstedt et al., 2020). A recent advancement in this field, is the concept of Smart Product-Service Systems (S-PSS), a term created to denote the integration of smart products and e-services into a unified solution through disruptive ICT (Valencia Cardona et al., 2014).

S-PSS employs digital technologies to transform business processes, enabling the delivery of customized offerings (Frank et al., 2019). The relationship between digitalization and servitization is a key element for the integration of digital offerings into S-PSS. This intersection brings attention to the role of digital platforms, which have evolved from intra-organizational information systems to become the backbone of digital ecosystems.

These digital platforms now serve as the infrastructure that facilitates how organizations interact in coordination, and collaboration within these ecosystems. They have also become important instruments in the provision of e-services to users. The platform approach, with its roles, elements, and modular architecture, is positioned as a way for organizations to deliver smart and customized products and services.

In this context, S-PSS not only employs digital technologies for the transformation of business processes but also requires the ability to capture the diverse needs of stakeholders and engage in co-creation for value generation. Furthermore, S-PSS aims to establish a lifelong partnership between suppliers and customers through the services offered, with User experience (UX) serving as a potential differentiator among competitors.

Recognizing the important role of digital platforms within S-PSS, it becomes notable that these platforms are tools for interaction and an important mean in which organizations deliver smart and interconnected services and allow users to interact with the physical world (via IoT devices, sensors, etc.).

Figure 1.1: Digital Platforms on the S-PSS ecosystem (a) Bosch (b) Lenze (c) Fitbit

1.1 Research Context

This thesis forms part of the Digital Manufacturing and Design (DimanD) Innovation and Training Network (ITN) programme—a multidisciplinary, multi-professional, and cross-sectorial European research and training network. As industries progressively transition towards servitization for more ecologically, economically, and socially sustainable solutions, it is important to consider human needs in this evolving landscape. Within DiManD, this research project aims to enhance the UX for various stakeholders. This is achieved by catering to the needs of the intended users, delivering them relevant and tailored information.

The UX of a S-PSS 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 (Figure 1.1). 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 of user experience within the S-PSS ecosystem. Therefore, user interfaces within these platforms play a fundamental role in shaping the UX.

User Interfaces (UIs) are platforms for cognition and communication between humans and devices, and they are the approach for information transmittal among user, products and services (Gong, 2009). The ability of interfaces to dynamically respond to user behavior, preferences, and changing contextual factors can be an effective method for optimizing UX. Adaptive interfaces within S-PSS can tailor the user experience in real-time, aligning product and service interactions with individual user needs. This adaptability can enhance user satisfaction and engagement, contributing significantly to the success of S-PSS implementations.

The contextual information plays a very important role in the UX and User-centred design (UCD). Context refers to any information relevant to a user task, as established by the ISO 9241-210 standard (Iso, 2010). By understanding and adapting to the context of use, which includes preferences and behaviors, it is possible to enhance the usability of a product or service and create a more tailored experience.

On the other hand, S-PSS are often characterized as "highly context-dependent" (Valencia Cardona et al., 2014). Context-awareness is the capacity of devices or systems to react to their environment and make decisions based on user situations. This ability can be used to enhance user experiences significantly by dynamically respond to changing contexts. This could be seen as a key aspect for creating adaptive and personalized UIs in S-PSS digital platforms.

1.2 Research Motivations

Despite the growing interest in S-PSS within the literature, the focus has predominantly been on the design stage. Numerous approaches exist to capture stakeholder requirements, utilizing various engineering methodologies or incorporating user data in the form of opinions during the usage stage. However, comprehensive studies on UX in S-PSS are scarce, often limited to the design phase and neglecting the digital, analytical, and connectivity capabilities inherent in S-PSS. Consequently, there is a need for frameworks or methodologies that facilitate the design of S-PSS, ensuring sustained UX enhancement throughout the entire life cycle. In a related study, Yu, Sung (2023) conducted an investigation into the adoption intention of S-PSS, with a specific emphasis on smart appliances. Their survey involved 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.

Hence, the primary motivation driving this research is to enhance the UX across the design and usage phases of S-PSS by leveraging context-awareness capabilities and the use of adaptive user interfaces. The emphasis lies in optimizing the UI of digital platforms used for service delivery, considering that S-PSS can present users with a multitude of services. Then, users may encounter an increased cognitive load due to the complexity of managing diverse functionalities (Kumar, Prajapati, 2019). Moreover, a higher visual complexity contributes to increased reaction time that may lead to user dissatisfaction with smart products and, consequently, diminish the behavioral intentions to continue using them

Another extrinsic motivations that underscore the significance of focusing on S-PSS are presented below:

- The surge in companies adopting servitization as a strategy, particularly in response to the economic challenges posed by the COVID-19 pandemic. The pandemic disrupted traditional manufacturing industries, leading to production and economic activity interruptions. A survey in Italy revealed that 66% of companies anticipated a significant reduction in product sales, while only 26% expected a similar decline in their sales of advanced services (Rapaccini et al., 2020) (Figure 1.2).

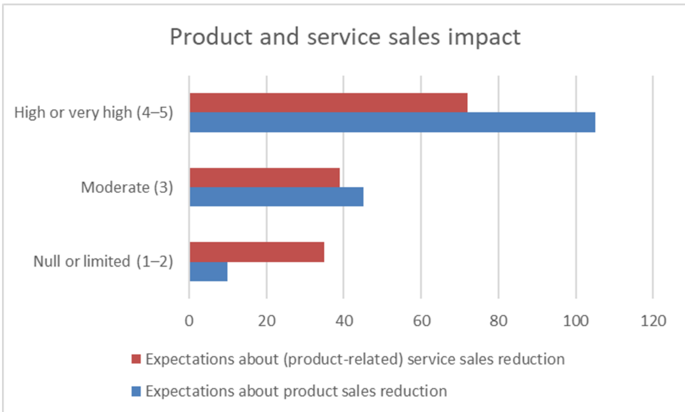


Figure 1.2: Perspectives on sales of services and products on COVID-19 pandemic. From Rapaccini et al. (2020)

- The potential for S-PSS to drive the Circular Economy—an economic system prioritizing reducing, reusing, recycling, and recovering materials and resources. S-PSS facilitates control over products at the end of their life cycle, as the relationship between customers and providers is built on the services they use. In certain scenarios, ownership of the product remains with the provider, necessitating designs that prioritize product durability and efficient processes.

- The increasing importance of customer satisfaction through personalization and customization. Modern customers expect products that adapt to their evolving needs, making it imperative to integrate these preferences into the design and delivery of S-PSS.

1.2.1 Thesis Objectives

The primary goal of this research is to develop a framework that integrates Adaptive User Interfaces (AUIs) into the design and utilization phases of S-PSS, with a specific focus on enhancing the overall UX. By leveraging context-awareness capabilities and integrating existing machine learning algorithms, the framework aims to provide methodological development guidelines for the design and implementation of AUI for S-PSS digital services. This involves tailoring the interfaces to dynamically respond to user behaviors, and contextual factors. Through the application of these guidelines, the thesis seeks to contribute to the development of reusable and effective solutions for the design challenges posed by the dynamic nature of S-PSS.

The main objectives of the research are summarized below:

- **Objective 1:** Develop a Framework for AUIs for S-PSS that leverages context-aware capability

  This objective involves the creation of a framework, as a tool to guide development of AUIs within S-PSS. The framework will emphasize context-awareness, by employing context-aware recommendation systems to promote adaptability to varying user needs and dynamic conditions.

- **Objective 2:** Validate the Framework Through Experimentation

  To validate the efficacy of the developed framework, an experimental phase will be undertaken. This will involve utilizing a smart device and associated services within a Business-to-Consumer (B2C) scenario. The experimentation aims to assess the practicality and effectiveness of the framework in real example and consumer-oriented settings.

- **Objective 3:** Implement Industrial Case Study

  This objective focuses on the application of the developed framework in Business-to-Business (B2B) settings, specifically within the context of servitization in industrial monitoring services. The goal is to adapt industrial user interfaces, ensuring seamless integration with S-PSS principles. The industrial case studies will provide insights into the framework's applicability and impact in complex, business-oriented scenarios.

1.3 Research Hypothesis

The hypotheses that this thesis seeks to demonstrate are the following:

- **Hypothesis 1**: The proposed framework results in a quantifiable positive impact on overall UX, demonstrated by a favorable user perception of ease of use and overall satisfaction.
- **Hypothesis 2**: The Context-Awareness capability enhances the quality of adaptations in the user interface of Smart Product-Service Systems during the *usage stage*, specifically in terms of accuracy, precision, and context compatibility of the adaptations.

1.4 Research Methodology

The methodology of this work is based on the Design science research methodology (DSRM) (Peffers et al., 2007). The ultimate aim of DSR is to solve a problem by creating an artifact (i.e. framework). The output of a DSR can be categorized as constructs, models, methods, instantiations (Peffers et al., 2007). This methodology has been used to define the stages of this research from the start and it has been useful as a roadmap in the elaboration of the research project, an overview of the methodology is depicted in Figure 1.3.

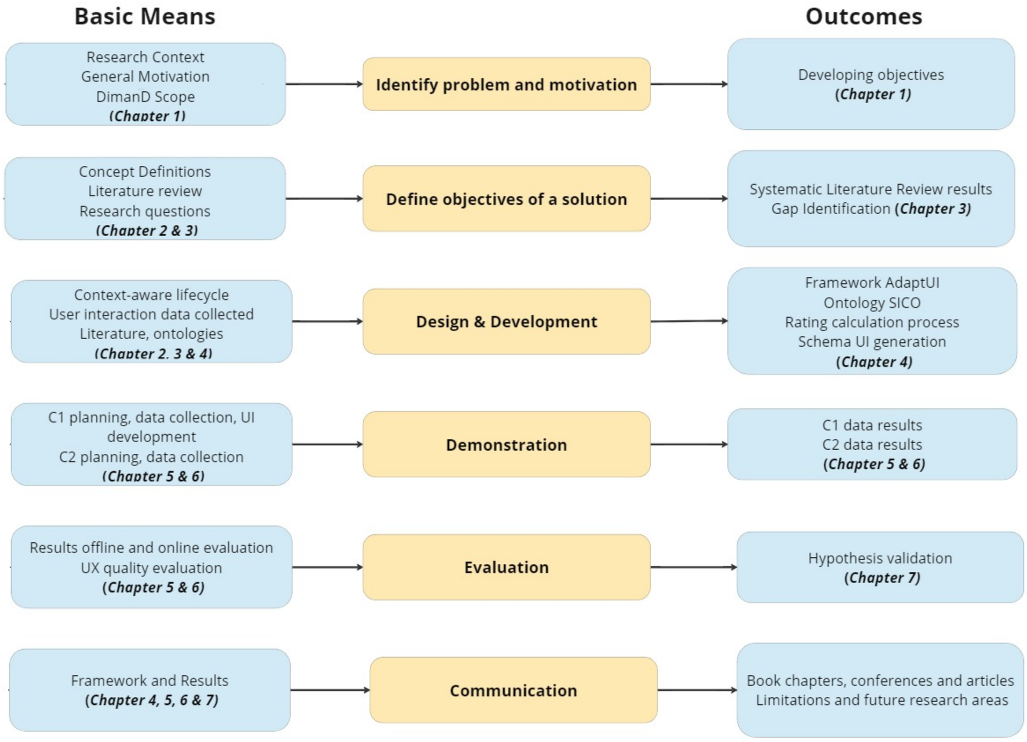


Figure 1.3: Design Research Methodology in the context of Information Systems

Step 1: Problem Identification and Motivation

- Set a general objective based on the DimanD research context and given motivations.
- Establish a general review on concepts and foundational background.
- Conduct a Systematic Literature Review (SLR), analyze existing design challenges within S-PSS.
- Explore UX issues and the need for context-aware interfaces.

Step 2: Define objectives of a solution and hypothesis

- Define the objectives of the research, based on research gaps and motivation.
- Aim to develop a Context-Aware framework for AUI in S-PSS.

- Specify desired outcomes, including improved user satisfaction, adaptability, and efficiency.

Step 3: Design & Development

- Development of framework based on context life cycle
- Design of an ontology model capable to represent features of S-PSS and relationships with contextual data and user interactions, based on user interaction data and other ontologies found on the literature..

Step 4: Demonstration

- Showcase the developed framework through practical demonstrations.
- Utilize real-world scenarios within Business-to-Consumer (B2C) and Business-to-Business (B2B) contexts.
- Emphasize the adaptability and effectiveness of the framework in enhancing user experiences.

Step 5: Evaluation

- Conduct evaluations, employing both quantitative and qualitative methods.
- Assess the framework impact on different aspects relevant to the UX such as, user satisfaction, ease of use and usability perception.
- Gather feedback from users in controlled experiments and real-world implementations.

Step 6: Communication

- Disseminate research findings through conference presentations, book chapters and journal publications.

1.5 Research Contributions

The main contribution of the thesis is the framework. This framework addresses the need to extend the analysis of the UX beyond the design stage, ensuring responsiveness to evolving user needs during the usage stage of S-PSS. Through the integration of ontologies and context-aware recommendation systems, with user interactions as the primary data source, the framework enables the development of AUIs that dynamically respond to user behaviour and contextual changes.

The significance of this contribution lies in the integration of various components, to create a practical and available tool. Through the presented case studies, the practical implementation of the framework is demonstrated, showing a hands-on development approach that considers technological aspects and appropriate tools. This empirical validation strengthens the framework practicality and effectiveness in real-world scenarios.

Below we summarize the main contributions of this thesis, the advances they present in the state of the art and the associated publications:

- The AdaptUI framework serves as a comprehensive tool for facilitating the development of adaptive user interfaces within the context of S-PSS, covering the entire process from design to deployment (Carrera-Rivera et al., 2024). At the core of the framework lies a Java-based API, which incorporates essential functionalities for seamless integration with ontology-related elements. The API also provides necessary endpoints for generating recommendations. Accessible at `https://github.com/aicarrera/contextawareUX4S-PSS`, the API codebase is openly available.
- The Service Interaction Context Ontology (SICO) is a fundamental component of the framework, serving to capture and model relationships among e-services, users, and contextual data, placing a specific emphasis on user interactions. SICO as ontology can help in the development of more effective and user-centered interfaces in service-oriented environments. SICO is available at `https://aicarrera.github.io/SICO/index-en.html`.
- A dataset of human-machine interactions, collected in a controlled and structured manner. The aim of this dataset is to support the development of adaptive Human-Machine Interfaces (HMIs) (Carrera-Rivera et al., 2023). Leveraging a custom-built web application developed in React framework with formally defined User Interfaces (UIs), the dataset underwent thorough processing and analysis to create a resource suitable for professionals and data analysts interested in user interface adaptations. Published in the Journal of Scientific Data (Q1), the corresponding paper includes open-source code and details of a replicable experiment, contributing valuable insights to the dynamic field of human-machine interfaces. This work enhances the accessibility and applicability of the contributions in the broader scientific community (available at `https://doi.org/10.6084/m9.figshare.c.6612805.v1`).

In addition to the contributions above, additional contributions of the thesis are:

- A comprehensive literature review of S-PSS forms the foundation for this thesis, adopting a systematic approach to explore context-awareness capability. The study includes a bibliometric analysis using a cluster keyword map, highlighting key topics in S-PSS, particularly in design and user experience. Additionally, an in-depth analysis of case studies aligns with the life cycle of context-aware applications. This exploration enhances the understanding of S-PSS implementations and establishes connections between design, user experience, and context-awareness. It is expected to guide future research on smart products and services, this contribution provides a discussion of future directions, finding and limitations for the field's ongoing development. The review was published in the journal Computers in Industry by Elsevier (Q1).

- Derived from the literature review, another contribution is a methodological guide designed for researchers, especially those in the early stages of their research journey, facilitating a systematic literature review in computer science. This guide outlines strategies for conducting such reviews and introduces an algorithmic approach to streamline the process, offering a valuable resource for researchers navigating this methodology. It was published in Methods X, Elsevier Journal (Q2).

Finally, Table 1.1 summarises the papers written to date (December, 2023).

Publication Title	**Sent to**	**Type**	**Status**
Context-awareness for the design of Smart-product service systems: Literature review. Carrera-Rivera, A., Larrinaga, F., & Lasa, G. (2022).	Computers in Industry	J (Q1)	Accepted (8/Jun/2022)
UX- for Smart-PSS: Towards a Context-Aware Framework	Computer-Human Interaction Research and Applications (CHIRA)	C	Accepted (8/Jun/2022)
How-to conduct a systematic literature review: A quick guide for computer science research	Methods X	J (Q2)	Accepted (September/2022)
Exploring the transformation of user interactions to Adaptive Human-Machine Interfaces	XXIII International Conference on Human Computer Interaction, Lleida, Spain, September 2023	C	Accepted (September, 2023)
Structured dataset of human-machine interactions enabling adaptive user interfaces	Scientific Data	J (Q1)	Accepted (November, 2023)
A context-aware framework for recommendation-based adaptive user interfaces for Smart Product-Service System	User Modeling and User-Adapted Interaction	J (Q1)	Second round revision (February, 2024)
A framework for the transformation of user interactions to Adaptive human-machine interfaces: A general overview	Future Perspectives on HCI Research - Springer HCI Series	BC	Publication pending (Feb,2024).
Others related to Dimand WPs			
Big data life cycle in shop-floor–trends and challenges	IEEE Access	J (Q1)	Accepted (March, 2023)
Towards autonomous manufacturing automation: Analysis of the requirements of self-x behaviours	Advanced Manufacturing	J (Q1)	In Revision (September, 2023)

Table 1.1: Description of Publications. Type **C** Conference **J** Journal **BC** Book Chapter

1.6 Thesis structure

The thesis is organized in chapters as follows. Chapter 2 sets the foundation by describing fundamental background of Servitization and S-PSS. It also addresses the life cycle of S-PSS and explores Context Awareness, covering aspects such as context types, and details the life-cycle of context-aware applications. Additionally, the chapter provides an overview of the concepts of AUI, including definitions, pillars, and methodologies, as well as frameworks found in the literature.

Moving on to Chapter 3, the state of the art is examined through a systematic literature review methodology. The chapter outlines research questions, and the approach for study selection and quality assessment. Additionally, it includes a bibliometric analysis to get insights from existing literature. The chapter then presents a discussion on user-centered design in S-PSS, context-awareness in S-PSS, and finally, it concludes by presenting opportunities for further research. The chapter is very relevant to the thesis, as it served as the foundation for the hypothesis formulated.

In Chapter 4, the thesis introduces the AdaptUI framework, a context-aware framework for AUIs in S-PSS. The generation of the AUI is achieved through the use of recommendation systems that leverage relevant usage data to predict services more pertinent to the user based on behavior. The chapter elaborates on each component of the framework, including acquisition, modeling, reasoning, UI adaptation, and monitoring and evaluation.

Chapter 5 presents a case study centered around a smart vending machine. The chapter begins with an introduction, followed by a detailed set-up of the case study, including a technical implementation overview. It explores the evaluation of the recommendation component, UX validation, and engages in a thorough discussion on the impact on performance, usability, user engagement, and satisfaction in S-PSS. The chapter concludes with findings and limitations.

Chapter 6 follows a similar structure to Chapter 5, presenting another case study that is focused on monitoring dashboards for industrial S-PSS. It begins with an introduction, outlines the case study set-up, provides technical implementation details, evaluates the recommendation component, and validates the UX.

Finally, Chapter 7 serves as the concluding section of the thesis. It provides a summary of findings, discusses the contributions to the field, and presents the validation of hypotheses. The chapter acknowledges limitations and suggests directions for future research.

2 Background

The objective of this chapter is to provide fundamental notions about central elements of the thesis. On one hand, the concepts of servitization and S-PSS are necessary for obtaining a comprehensive overview that enables an understanding of why the framework is oriented towards these digital service platform ecosystems. On the other hand, the description of the capability of context-awareness and the elements of context-aware systems, in conjunction with Adaptive User Interface (AUI), facilitate the understanding of the subsequent chapters.

2.1 Servitization and Product Service Systems

First defined in 1988, “Servitization” describes a shift in focus in which companies add value to their core offerings through services as a way to improve competitiveness (Vandermerwe, Rada, 1988).

Product-Service Systems (PSS) is a strategy to integrate products with specificly related services to satisfy user needs (Tukker, 2004). PSS are seen as a way to simultaneously maximize the value of the product and increase profits (Hallstedt et al., 2020). However, this strategy present multiple challenges, such as business implementation complexity, technology gaps, and industry maturity.

From the business perspective, PSS can be classified into three main categories (Tukker, 2004):

- **Product-oriented**
  This category refers to services that facilitate the sale of products, add functionality, and/or personalize existing products. Examples include warranties, technical support, instructional apps, and monitorization. In this case, ownership is transferred to the customer.
- **Result-oriented**
  The customer pays for a result or performance, rather than a product per se. The provider

is thus free as to how to deliver the result. For instance, a service that provides "good quality air". The ownership usually remains with the provider, and the customer pays a fee for the service. Tukker (2004), argued that result-oriented services present the greatest potential for reducing the impact of resources as compared to other types of PSS. Since the concept of selling a "result" is quite abstract, the provider will attempt to do so in the most cost-effective way, which can lead to radical innovations. By way of illustration, large companies may offer product-as-a-service options in some markets, whilst retaining the standard product sales model in others. One such example is Philips, the globally renowned technology company focused on electronic and health devices. To diversify the business, Philips offers "Light as Service" directed towards the market of offices and buildings. The service delivers a complete lighting solution, with the design of a customized lighting plan, delivery and installation of lightning, and maintenance and monitoring included.

- **Use-oriented**
  Here, the customer pays for usage of the product and ownership remains in the hands of the provider. Three main scenarios are described in the literature: leasing, renting, and product pooling, in which several customers use the product simultaneously. A prime example is the case of BMW and Daimler. Both companies had their own car-sharing solutions (pay-per-use), and shortly after merging formed ShareNow, a service which offers free parking, fuel, and insurance. Customers benefit from high availability and paying only for the time they use the vehicle. At the time of writing, operations are centred in European cities. Attempts were made to position themselves in North America, but ultimately abandoned due to 'low adoption rates' (Hawkins, 2019)

The advancement of digital technologies has led to the emergence of innovative business models capitalizing on the digital capabilities of Cyber-Physical System (CPS), which can be defined as systems that integrate physical processes with computational capabilities. This technological shift prompts a reconsideration of traditional PSS, giving rise to the introduction of S-PSS.

2.2 Smart Product-Service Systems (S-PSS)

Digitalization is the use of digital technologies to transform business processes (Frank et al., 2019) and is the enabler of Smart Product-Service Systems (S-PSS). Digital technologies can be used to deliver smart and interconnected services that can also interact with the physical elements such as IoT devices and sensors to perform tasks including data analytics and big data processing. S-PSS have been defined by several authors , who have classified key aspects such as relationships with multiple stakeholders, smart products, and related digital services (see Table 2.1). In the present study, we highlight the definition of Zheng et al. (2019b) *"(S-PSS) is an IT-driven value co-creation business strategy consisting of various stakeholders such as players, intelligent*

Definitions	References
"Smart products and its generated e-services into a single solution by embracing disruptive ICT."	(Valencia Cardona et al., 2014)
"A digital-based ecosystem of value creation characterized by high complexity, dynamics and interconnectedness among stakeholders."	(Kuhlenkötter et al., 2017)
"A platform service ecosystem, in which the platform is made up of smart products and smart services, while multiple service systems constitute a service ecosystem."	(Liu et al., 2018)
"Digital-enabled business solution supplied within an ecosystem which provides economic and sustainable value to the customer by integrating into a unique offer intelligent products with data-enabled services allowed by physical and digital infrastructures"	(Pirola et al., 2020)
"S-PSS could be regarded as cyber-physical products (CPP) which are described as a physical product that offers ICT-enabled services via either a built-in or an external network connecting device, allowing users to use the product functionality and communicate information during its use phase for acquiring complementary service support"	(Chou, 2021)

Table 2.1: Definitions S-PSS

systems such as infrastructure, smart, connected products such as media and tools, and their generated e-services such as key values delivered that continuously strive to meet individual customer needs in a sustainable manner".

Therefore, S-PSS is the convergence between products with high levels of digitalization that serve business processes and related services that are customer-oriented. Zheng et al. (2019b) identify digital capabilities as one of the key factors that differentiate S-PSS from traditional PSS (figure 2.1). The first step towards product digitization starts with intelligence capabilities, embedded systems and sensors are integrated into hardware to capture data and facilitate interaction between the machine and its environment. Then, the ubiquitous connectivity of smart products delivers IoT technology and wireless communications, which are fundamental for service providers to collect data. Data processing is further improved with cloud computing, which allows multiple devices to interact with each other. The final link in the chain is analytic capability, converting the available data into valuable insights and actionable directions for businesses (Lenka et al., 2017).

These technological advancements serve as enablers for realizing the core business foundations of traditional PSS: Product-oriented, Result-oriented, and Use-oriented.

2.2.1 S-PSS Life Cycle

It is possible to analyze S-PSS using the existing traditional product life cycle, with certain key differences. The product life cycle is divided into three main stages (Kiritsis, 2011): the beginning of life (BOL), middle life (MOL), and end of life (EOL). The beginning of life starts

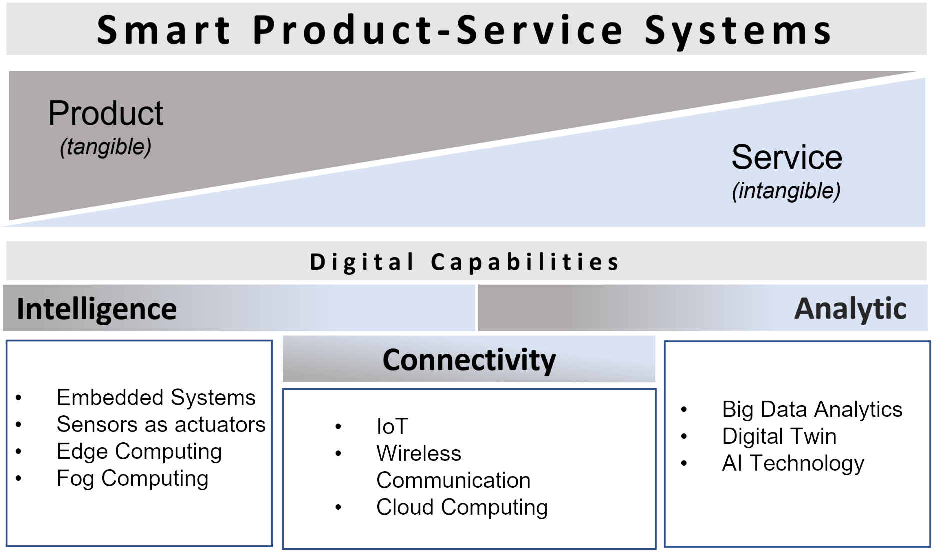


Figure 2.1: Smart Product Service System Graphic Description

when a product is designed according to requirements and then manufactured. The middle of life is related to the use, service, and repair of the product. Finally, end-of-life represent the approaches to reverse logistics, for instance, recycling, reuse, re-manufacture, or disposal.

In S-PSS the design stage begins with the requirement elicitation process, where customer demands are identified. However, in contrast with the traditional process, designers can rely on data from user-behaviors and demands that can be also obtained in the manufacturing and usage stages. Services can be classified as product-independent services (e-services) and digitalized services that are dependent on the smart capabilities of the physical product, such as monitoring, control, optimization, and autonomy. In the case of digitalized services, digital twins or augmented reality are proving to be a game changer in facilitating more accurate decision making (Zheng et al., 2019b).

The usage stage differs significantly from traditional PSS due to the smart capabilities of connected products and related services. Starting with smart operation and maintenance, which involves assessing the performance of PSS and monitoring the operations. Plus, smart reconfiguration which is the modification of a product to fulfill new requirements, enabling it to react to a new context.

Finally, End of life (EoL) is characterized by various scenarios such as reuse of the product with refurbishing or re-manufacturing, recycling, and disposal. In traditional product sales, the information loops are cut after the sale. In contrast, S-PSS can reclaim the information, and bring data to manufacturers and designers (Kiritsis, 2011). Alcayaga et al. (2019) refer to this as the “Smart-circular system”, integrating the terms “Smart-PSS” and “circular economy”. The new capabilities delivered by S-PSS demonstrate great potential to improve processes that can lead to a better use of resources at end of life. As previously described, this new perspective on the product life cycle does not necessarily change application or interpretation of the phases, but rather highlights the development of S-PSS as an incremental process. In contrast with the traditional product life cycle, where each phase is clearly divided, in S-PSS is necessary to take a holistic approach, taking into consideration the fact that each phase is interconnected with the others (Wuest, Wellsandt, 2016).

2.3 Context Awareness

Context was defined by Abowd et al. (1999) as *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*. Therefore, raw data extracted from multiple sources does not by itself provide context. By way of illustration, the data generated by the accelerometer of a mobile phone—after being processed—can provide information about the user, such as the activity they were engaged in at any given time.

In general, a context-aware system uses context to deliver information and/or services relevant to the task of a user (Abowd et al., 1999). Thus, a context-aware system does not necessarily imply automatization or real-time processing, instead, it refers to the ability to respond to context. However, as context does not come from raw data, there must be a process of interpretation and detection. There are three main approaches for the development of context-aware systems (Hu et al., 2008):

- No application-level context model: The application communicates directly with context data sources (i.e. sensor, machines, etc.) and processes the data to make decisions.
- Implicit context model: Libraries or toolkits are used to process context information.
- Explicit context-model: A middleware application is created to manage data gathering and processing using a well-established context model.

The third approach is more scalable and multiple context-aware applications can be connected to the middleware. This approach also allows decentralization and distribution by connecting these services in the cloud. Context-aware applications can perform on mobile devices, desktop, or web environments. In fact, context has already become a type of service, Context-as-a-Service (CXaaS) (Hynes et al., 2009).

2.3.1 Context Types and Categorisation

According to Perera et al. (2013), context can be divided into two perspectives, *Operational* and *Conceptual*. *Operational* categorises context based on how it is acquired, modelled, and treated, as set out in Table 2.2.

Conceptual categorises context based on meaning and relationships. Abowd et al. (Abowd et al., 1999) introduced one of the first categorizations, identifying four main groups: location,

Operational Context	Description
Primary	Any information retrieved without any kind of pre-processing or additional operations. (i.e temperature obtained directly from sensors)
Secondary	Any information that is obtained from the primary context. Can be computed using sensor data relationships.(i.e average temperature of a building obtained from multiple sensor readings)

Table 2.2: Context 'Operational' categorization

Conceptual/ Operational	Identity	Location	Time	Activity
Primary	User identity based on NFC tag	Location from GPS data	Read timestamp from device	Data from user interactions
Secondary	Retrieve friend list from user based on profile	Obtain places close to a location	Infer the season based on the time	Identify most visited apps based on interactions

Table 2.3: Context categorization based on Conceptual and Operational perspectives

identity, time, and activity. Table 2.3 presents examples from both the operational and conceptual perspective. Several scholars have since added to the classification: user, computing, physical, social, networking, things, sensors and contexts. It is the opinion of the author that these are sub-categories from the original Abowd classification, however they facilitate more meaningful and structured organisation of the context data.

There is no only one approach that can be used in a context-aware system and it would need the combination of different approaches. The need for combining multiple approaches in a context-aware system arises from the diversity of contextual information. Context, in various applications, is varied and subject to constant change, involving factors such as location, user preferences, and environmental conditions.

2.3.2 Context Life Cycle

Perera et al. (2013) defined four general steps in the context life cycle as a way to describe the process of developing context-aware applications. Figure 2.2 presents the context life cycle, with the addition of a *Monitoring* phase to support continuous evolution of the application. The first step is *Context Acquisition*, which refers to the data to be acquired from different sources (i.e. sensors, CPS, databases, etc). In Context modeling, the collected data must be represented in a meaningful way. Then, in Context Reasoning, data is processed to provide useful information and insights (context). The Context Dissemination phase distributes context to consumers, which can be end-users or other applications. Finally, Context Monitoring is a stage not broadly represented, but should be considered after dissemination because the context may change at some point. Systems or applications have to be able to identify changes and

Phase	Techniques
Acquisition: Obtaining data from sensors	Event based By type of sensors: physical, logical, virtual By Acquisition process.
Modeling: Representing data in meaningful units	Key-Value Object Based Ontology Based
Reasoning: Deducing new information from multiple data sources.	Supervised Learning Unsupervised Learning Ontology Based Fuzzy Logic Neural Networks
Dissemination: Delivering results to users	Query Based Publish and subscribe

Table 2.4: Context Life Cycle

update the models accordingly. Table 2.4 details the techniques used in each of the stages, that are described in the following subsections.

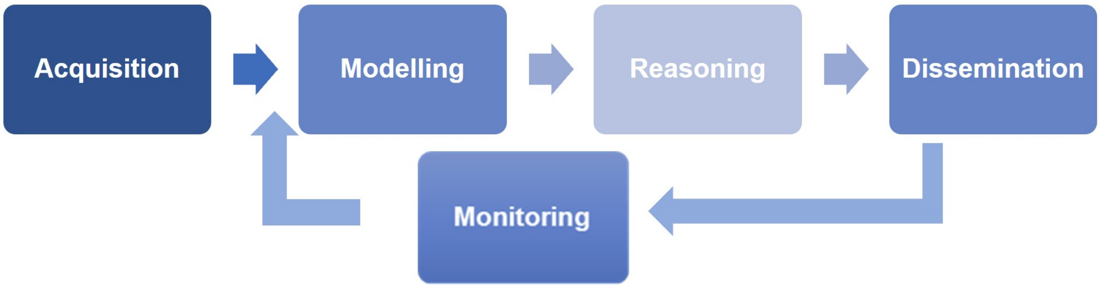


Figure 2.2: Context Life Cycle

2.3.3 Context acquisition

Perera et al. (2013) have described in detail the multiple methods of data acquisition based on five factors: responsibility, frequency, context source, sensor type, and acquisition process.

Most of the data used in context-aware systems is acquired through physical sensors, however, a sensor refers to any data source that provides relevant context. Table 2.5 describes three types of sensors: physical, logical, and virtual (Perera et al., 2013). The context source for physical sensors can be different, depending on scalability and applicability.

In Table 2.6, responsibility, frequency of events, and acquisition process factors are described. Each of these factors should be considered independent from sensor types and multiple techniques might be needed to work together. For instance, if a sensor detects an instant event like ´turning on a light´, the event data might be pushed to the software component, and thus responsibility and frequency of events should be considered.

Sensor Type	Description
Virtual	No physical sensors. Retrieve data from applications i.e. calendars, emails, etc.
Logical	Software sensors, combine physical and virtual sensors.
Physical	Tangible sensors • *Direct from hardware* • *From Middleware* • *From Context Server*

Table 2.5: Sensor Types

Responsibility	**Pull**: The software component responsible for acquiring sensor data, makes a request (i.e., query) from the sensor periodically(i.e., after certain intervals) or instantly to acquire data.
	Push: The physical or virtual sensor pushes data to the component that is responsible for acquiring sensor data periodically or instantly. Periodical or instant pushing can be used to facilitate a publish and subscribe model.
Frequency Events	**Instant**: Instant events are triggered by actions (i.e., opening a door). Sensor data is obtained at the moment the event occurs.
	Interval: sensors periodically monitor for a specific event to occur(i.e., snow, rain). Sensor data needs to be acquired for a certain period of time.
Acquisition Process	**Sense**: The data is obtained through sensors, including data stored in databases.
	Derive: The data is generated by performing a mathematical function or a computational operation. (i.e., acceleration from accelerometers)
	Manually: Users can provide context data through software, e.g., preference settings in tasks or configuring notifications.

Table 2.6: Context acquisition factors

2.3.4 Context modeling

Context modeling refers to the process of representing data in meaningful elements. This method typically involves the analysis of multiple data sources, to define entities (distinct objects or elements within the system), properties (attributes or characteristics of these entities), relationships (connections or associations between entities) and dependencies (Bettini et al., 2010). In executing this process, it is important to acknowledge, that the quality of data might vary significantly between sources. Sensor accuracy, for instance, is affected by multiple factors such as environmental aspects, incorrect installation, and lack of maintenance. Subsequently, a data pre-processing serves as an intermediate step between the acquisition and modeling stages, ensuring the correct instantiation of entities within the model.

Table 2.7 presents some of the most widely-used techniques. This study employs ontology-based modeling, which has the benefits of providing a structured and standardized approach for knowledge representation. However, a clear differentiator is the ability to provide a more precise and explicit representation of concepts and relationships within a domain. This will be described in greater detail in the following section.

Ontology-based modeling

Ontologies are a means to formally model concepts from a particular domain into a detailed specification of entities with properties and relationships (Gruber, 1993)(Guarino et al., 2009). They are created to manage heterogeneity and interoperability issues, by providing a shared view of a conceptualization. Web Ontology Language (OWL), a semantic language developed by W3C, is the formal way to express ontologies. It features a number of syntaxes including RDF/XML, OWL/XML, and Turtle. OWL is strongly influenced by Description Logic (DL), and depending on the version, can exploit reasoner implementations and make inferences from the developed model and data (Horrocks, 2005).

One of the most compelling reasons for the use of ontologies is their ability to share a common understanding between people, knowledge-domain experts, and software agents. Guarino (1998) first classified ontologies for their level of generality as a means to deal with the problem of integrating multiple ontologies. They considered that it is more appropriate to rely on top-level ontologies rather than merely trusting the agreements based on intersections between them.

From the highest level of generality, top-level ontologies are used to describe very general concepts not related to a particular domain, i.e., time, objects, matter, etc. An example of a top-level ontology is the Basic Formal Ontology (BFO)(Spear et al., 2016), primarily designed to specify material or immaterial entities in the scientific domain (biomedicine). Its use has since been extended to other scopes (e.g., agriculture, military), to create more specialized ontologies.

Domain ontologies define the vocabulary related to a particular domain. Extensive work has been carried out to develop domain ontologies in several areas of knowledge such as medicine, mechanics, law, and chemistry (Guarino et al., 2009). The Semantic Sensor Networks ontology (SSN) (Compton et al., 2011) is an example of a domain ontology, containing concepts and relationships related only to sensors. Concepts connected to other domains can be included from other ontologies when the ontology is deployed. Hence, SSN is a single-subject ontology that allows modularity and reusability.

Task ontologies are at a lesser level of generality, and aim to formalize knowledge for problem-solving processes in a specific task (e.g., selling a car, planning maintenance, etc) (Mizoguchi et al., 1995). This knowledge can be divided into two facets: task decomposition

Techniques	Description
Key-value	Data is represented as key-value pairs which can be presented in text format or binary files. ***Pros:*** Easily manages small amounts of information. ***Cons:*** Difficulty representing complex models and relationships.
Markup scheme	Data is modelled using tags. There are several markup languages e.g., XML, JSON. ***Pros:*** Allow efficient transfer of information and model data with specific schemes. ***Cons:*** Application dependent. Complex to model with many levels of information.
Object based	Data is modeled using class units, hierarchies and relationships. Object Oriented Programming (OOP) promotes reusability, abstraction and encapsulation. ***Pros***: Easy integration with OOP programming languages. Allows relationship between models. ***Cons*** Application dependent. Depend on principles but not standards.
Graphical modeling – Databases	Use UML or ORM that can easily be translated to SQL or NO SQL databases. ***Pros*** Allow relationships between models. Easier data retrieval. Validation through constraints. ***Cons*** Interoperability between different platforms is complex. Configuration required.
Ontology based	Data is modelled using ontologies, some standards can be used e.g., (RDF, RDFS, OWL) *Pros*: Allows semantic reasoning. *Cons*: Complex to create.

Table 2.7: Context modeling Techniques

and knowledge roles. Task decomposition divides the task into multiple subtasks, each with its specific goal, and also defines the control flow. The knowledge roles facet specifies the concepts and relationships of the task (Martins, Almeida Falbo de, 2008). In contrast with Domain Ontologies, Task Ontologies are not always developed to a level of high reusability. Thus, it is key to ensure domain knowledge reusability across all tasks.

Finally, at the lowest level of generality, application ontologies are the integration of Domain and Task Ontologies and are often specializations of both.

The levels of generality that ontologies provide help to hierarchize knowledge from domain to operational, facilitating the interoperation of multiple interrelated components in CPS and IoT solutions.

2.3.5 Context Reasoning

Context reasoning is the process of obtaining new information from multiple context-data sources that is useful for a task or user. Context reasoning can be divided into three steps (Perera et al., 2013) as depicted in Figure 2.3.

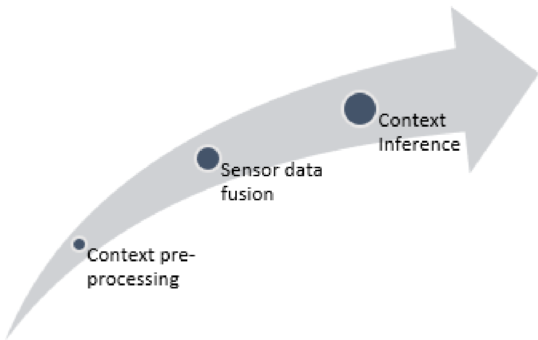


Figure 2.3: Context Reasoning Steps

1. Context pre-processing: tasks related to this step arise from the imperfect nature of data received. Data needs to be cleaned, outliers removed, and missing values completed.

2. Sensor data fusion: multiple sources of data need to be merged to produce more accurate outcomes. For instance, CPS are especially designed and created with multiple sensors that produce data.

3. Context Inference: the most important step which represents the transformation from "low-level" or raw data to "high-level" context. New information is derived from reasoning or machine learning techniques.

There exist multiple techniques to process context-reasoning. Table 2.8 sets out the most

Technique	Complexity	Description	Related Work
Rule Based	Low	Rules are structured in an IF-THEN-ELSE format. Prone to errors, rules are manual.	(Alexopoulos et al., 2018)
Fuzzy Logic	Low	Different from Boolean logic, considers intermediate values between 0 and 1.	(Araújo et al., 2018; Pandya et al., 2020; Kayes et al., 2017)
Ontology Based	Medium	Based on Description Logic: Maximizes the benefits of Ontological modeling. Cannot manage missing data or ambiguous information.	(Kayes et al., 2017; Goel et al., 2017)
Probabilistic Logic	Medium	Makes decisions based on probabilities. Often used to understand event occurrence. Limited to numerical values only.	(Goel et al., 2017; Kordestani et al., 2019)
Supervised Learning	Complex	Can make predictions using labelled data. Multiple well-known algorithms (e.g., Decision trees, SVM, KNN). Requires large amount of information to train models.	(Gross et al., 2017; Pan et al., 2019)
Unsupervised Learning	Complex	Groups unlabelled data into clusters. Can find patterns. Results are unpredictable.	

Table 2.8: Context Reasoning techniques

common together with their level of complexity, characteristics, and related work in the context of IoT and CPS (not necessarily related to industry 4.0).

2.3.6 Context Dissemination and Monitoring

Context Dissemination refers to the methods employed to deliver context to clients or users. From the customer or user perspective, it involves the process through which they acquire results from the "reasoning" process. The factors of responsibility and frequency presented in Table 2.6 during the acquisition stage are also useful in this process and should be considered. For instance, responsibility can manifest as a "Pull" mechanism where a software component periodically queries sensors for data, or as a "Push" mechanism where sensors actively transmit data at intervals (Perera et al., 2013).The frequency of events can be instantaneous, triggered by specific actions, or interval-based.

Furthermore, two distinct methods identified by (Perera et al., 2013) contribute to context dissemination:

- Query: Context consumers make a request and the system utilizes the request or query to produce results.
- Subscription: Context consumers subscribe to the system. The system returns results

periodically or when an event occurs (i.e., some threshold is met).

Context Monitoring is another stage directly related to Dissemination and Reasoning, considering that the context may evolve over time. Systems or applications must be able to identify changes and update the models used for Reasoning stage accordingly. Equally important is monitoring user feedback regarding the decisions made by the application.

Feedback from users can be obtained from integrated surveys, but also from more subtle approaches. For instance, if a user is not happy with the decision, service or information that the application is offering, they are likely to return to an old state of the application, not interact with the information, or not use the option, all of which can be interpreted as negative feedback. Reinforcement learning can be used to change the behaviour of the system. (Nurmi, Floréen, 2004).

2.4 Adaptive User Interface (AUI)

An Adaptive User Interface (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, 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 2.9. 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 on Figure 2.4.

The landscape of AI encompasses a diverse array of tools, techniques, and algorithms. In the development of AUIs, various AI approaches come into play. An early example is the "AVANTI" system presented by Stephanidis et al. (1998). This system introduced a conceptual framework aimed at dynamically adjusting the user interface 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

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, Rocha da, 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)

Table 2.9: Definitions Adaptive User Interface (AUI)

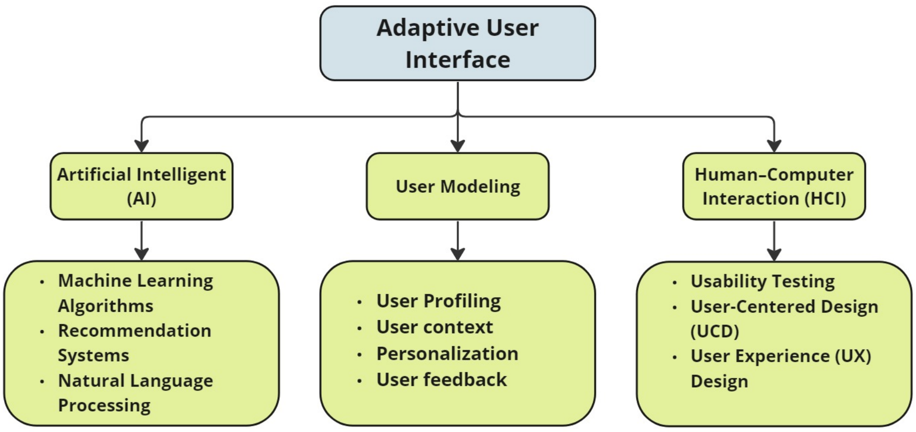


Figure 2.4: Adaptive User Interface pillars

an enhanced and tailored experience.

Similarly, Khan, 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 Systems (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.

Study	Solution offered	Research Type	Domain	Intelligent entity	UX	Usability	Method
Stefanidi et al. (2022)	M	Evaluation research	Software	Interface	Yes	No	Expert based evaluation, User testing
Gil et al. (2021)	F	Evaluation research	Software engineering	Interface	No	Yes	User testing, Questionnaire
Abrahão et al. (2021)	F	Solution proposal	Human-computer	Interface	N/A	N/A	
Berman et al. (2021)	F	Validation research	3D printing	Interface	No	Yes	User testing, Questionnaire
Gan et al. (2020)	F	Validation research	Human-computer	Other model	N/A	N/A	Interaction research
Johnston et al. (2020)	F	Validation research	Software engineering	Interface	Yes	No	Automated testing research
Johnston et al. (2019)	F	Solution proposal	Software engineering	Interface	N/A	No	
Stumpf (2019)	F	Solution proposal	Human-computer	System	No	No	Interaction research
Tan et al. (2018)	F	Validation research	Software engineering	System	No	No	Interaction research
Landowska et al. (2016)	M	Validation research	Software engineering	System	N/A	N/A	
Zhou et al. (2015)	F	Validation research	Human-computer	N/A	N/A	N/A	
Peck et al. (2015)	F	Solution proposal	Software engineering	System	No	No	
Fernandez-Garcia et al. (2015)	M	Solution proposal	Software engineering	Interface	No	No	
Mezhoudi et al. (2015)	M	Validation research	Human-computer	Interface	No	Yes	Interaction research

Table 2.10: Frameworks (F) and Methodologies (M) published in the Adaptive User Interface (AUI) literature

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. From the 212 studies reviewed, only ten corresponded to frameworks and four to methodologies (Table 2.10) (Brdnik et al., 2022). 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.

Table 2.10, nine of the fourteen described studies do not include any kind of UX or usability evaluation. These studies correspond to "solution proposals" or "validation research". The remaining five 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.

AUIs can offer multiple advantages to S-PSS. S-PSS integrates physical products with digital services, offering comprehensive solutions to users, and in this context AUIs can provide personalization, individually tailored instructions, and assistance. Such adaptations not only elevate user performance but also enhance the overall UX and acceptance of the system.

2.5 Conclusion

This chapter introduces foundational concepts for the thesis, providing a necessary background for the subsequent chapters. Servitization signifies a paradigm shift for companies, enhancing the value of their core offerings through services. Within this landscape, PSS emerges as a unified value proposition combining products and services. With the advancements of IoT technologies and the rise of CPS, a new term originates—S-PSS, denoting products with advanced digital capabilities that offer innovative services.

The life cycle of S-PSS is discussed in this chapter to better understand its evolving nature. In view of this, the assertion is made that context-awareness is a capability needed in S-PSS. Each stage of the context-aware life cycle is thoroughly examined, starting with the acquisition of data from smart devices, followed by the modeling of this information, reasoning, dissemination, and monitoring.

Furthermore, this section sets out the fundamental background relevant to the creation of the framework presented in this thesis, in particular, AUIs. Employing AI techniques, these interfaces adapt in real-time to provide personalization and customization to users.

The next chapter presents a Systematic Literature Review (SLR), allowing an understanding of the relationship between S-PSS, context-awareness, and their potential for enhancing the UX.

3

State of the Art

In recent years, there has been marked increase in the exploration and development of S-PSS, particularly centred on the design stage. Various methodologies, such as TRIZ, Quality Function Deployment (QFD), Kansei engineering (KE), Axiomatic Design (AD), Blueprint Design, and adaptable design, have been proposed to capture stakeholder requirements and enhance the design process (Cong et al., 2020a; Chou, 2021; Cong et al., 2020b). However, while these efforts have primarily concentrated on design aspects, the effective integration of user perspectives during usage remains a challenge.

To address this gap, some authors have worked to leverage user data and opinions from external sources or third-party applications to enhance the adaptability of S-PSS (Wang et al., 2020). In contrast, the field of User-Centered Design (UCD) endeavors to understand and empathize with potential users from the very conception of S-PSS, and aims to proactively address user needs and improve the overall UX.

As the design of S-PSS involves multiple intricate elements, it was crucial to establish a research focus that not only identifies existing gaps and challenges but also explores the potential of incorporating "context-aware" capabilities. This requires extending the field of study beyond the design stage, delving into how context-awareness could significantly impact the UX during usage, and exploring ways to exploit internal data sources within the S-PSS. Consequently, the present chapter seeks to contribute to this ever-changing landscape by investigating the role of context-awareness within S-PSS, with a particular focus on the UX in the usage stage.

Furthermore, this chapter represents the foundation and origin for the hypothesis, goals and objectives set on this thesis (Chapter 1).

3.1 Literature Review Methodology

A Systematic Literature Review (SLR) is used to identify, evaluate and summarize the research relevant to a particular group of research questions. It can be used to identify any gaps in the literature and suggest areas for further investigation (Keele, others, 2007). Parsif.al [1] is an online tool that supports researchers in developing an SLR, especially in the context of Software Engineering (Simple Complex, 2021). It was used in the present study to plan and manage the several stages of the SLR (Planning, Study Selection, Quality Assessment, and Data Extraction).

The goals of this review were as follows:

1. Explore the state of the art in the design and UX of S-PSS
2. Explore the state of the art of *Context-Awareness* approaches for the design and UX of S-PSS

Figure 3.1 depicts the protocol employed in the SLR. In the initial planning phase, keywords are carefully selected to form research questions. Digital libraries are then chosen for information retrieval, and the selected keywords are used to create queries. Simultaneously, inclusion and exclusion criteria are defined to screen papers, complemented by a quality assessment checklist. These artifacts are used to conduct the review , involving the gathering of studies while filtering out papers that do not adhere to the predetermined inclusion and quality criteria.

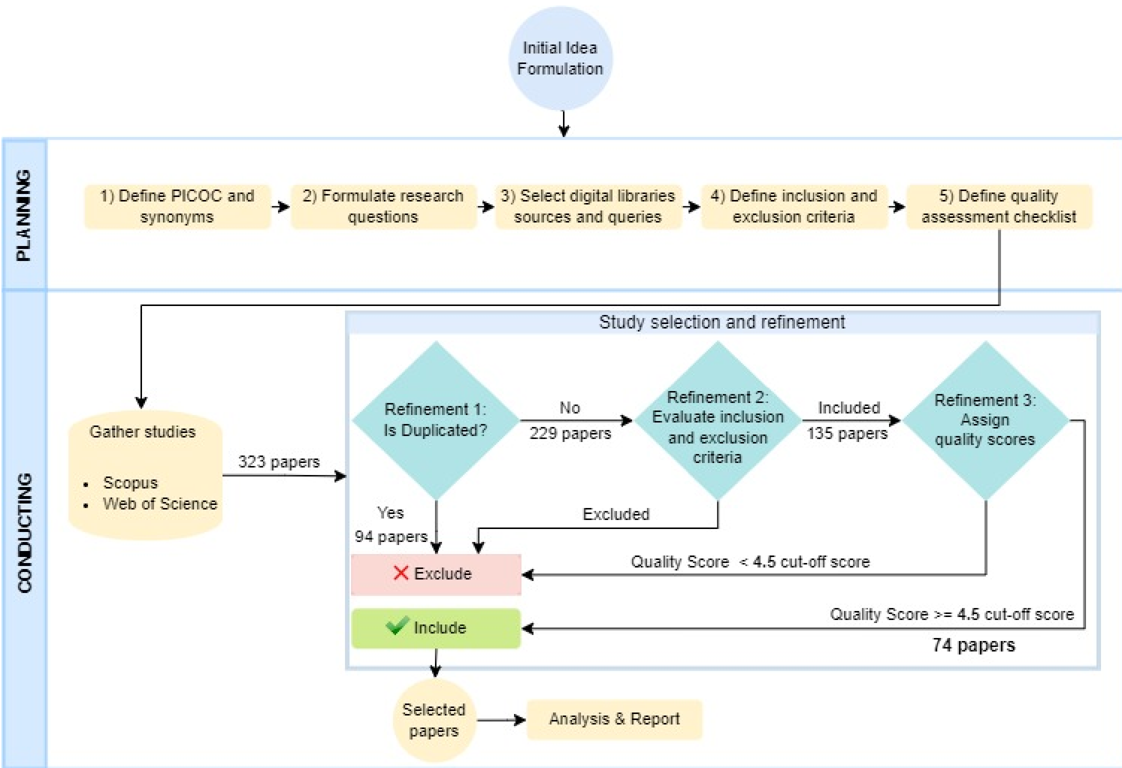


Figure 3.1: Systematic Literature Review (SLR) methodology

[1] https://parsif.al/

3.1.1 PICO and Research questions definition

Research questions set the focus for identifying relevant studies and data extraction (Wohlin et al., 2012). The PICO (Population, Intervention, Comparison and Outcome) model was used to break down the objectives of the review into searchable keywords and help formulate research questions (Petersen et al., 2015). PICO has been broadly used in medical and social sciences, to encourage researchers to consider the components of the questions (Petticrew, Roberts, 2008). Keele, others (2007) provided a detailed guide for SLR in the context of software engineering, establishing a correspondence of PICO elements. Table 3.1 describes the elements and their use in this review.

Population	Application area or Industry domain.	Smart-Product Service Systems.
Intervention	Methodology/tool/technology that addresses an specific issue.	*Context-awareness* capability.
Comparison	Methodology/tool/technology in which the intervention is being compared.	–Not applicable–
Outcome	Factors of importance to practitioners.	User-Centric Design and UX of S-PSS

Table 3.1: PICO Criteria

Three questions were formulated based on the PICO criteria, the definition of S-PSS, and the objectives established.

(RQ1): What relationships exist between UCD, UX, and Context-Awareness in the context of S-PSS?

(RQ2): How has *context-awareness* capability been used in the design of S-PSS?

(RQ3): What are the current gaps and challenges in the design of S-PSS to satisfy user needs and improve UX?

3.1.2 Digital Libraries selection and Search query string

Keywords and synonyms were chosen based on the PICO criteria, including the following main terms: *Smart Product-Service Systems*, *Context awareness*, *Service Design*, and *User Experience*. Each set of searches was conducted on the databases: Scopus and Web of Science, which were selected for their advanced search capabilities and because they deliver the broadest interdisciplinary content. The search strings used for each database can be found in Table 3.2, alongside the number of results. Originally, this SLR (Carrera-Rivera et al., 2022) presented results until December 2021 but has been updated to January 2024.

The chosen keywords were applied to all titles, abstracts, and metadata. Parsif.al was utilized during this stage to efficiently handle any potential duplicates.

Database	Query	Nº Results
Scopus	`TITLE-ABS-KEY ('SMART PRODUCT SERVICE*' OR 'SMART SERVICE SYSTEM*' OR 'PRODUCT SERVICE SYSTEM') AND PUBYEAR > 2015 AND (TITLE-ABS-KEY ('CONTEXT AWARE*') OR TITLE-ABS-KEY ('SERVICE DESIGN' OR 'USER EXPERIENCE' OR 'USABILITY' OR 'USER CENTRED DESIGN')) AND (LIMIT-TO (DOCTYPE,'ar') OR LIMIT-TO (DOCTYPE,'cp')) AND (LIMIT-TO (LANGUAGE,'English'))`	212
WoS	`(TS= ( 'SMART PRODUCT SERVICE*' OR 'SMART SERVICE SYSTEM*' OR 'PRODUCT SERVICE SYSTEM' ) AND ( TS= ( 'CONTEXT AWARE*' ) OR TS= ( 'SERVICE DESIGN' OR 'USER CENTRED DESIGN' OR 'USER EXPERIENCE' OR 'USABILITY' ) ))`	111

Table 3.2: Query and query results by database

3.1.3 Study selection: Inclusion and Exclusion criteria and Quality Assesment

To ensure effective selection, articles were excluded based on titles and abstracts, following the exclusion criteria set out in Table 3.3. When in doubt, the document was skimmed and accepted or rejected in accordance with these criteria.

Inclusion Criteria	Exclusion Criteria
• Articles that include at least 2 of the keywords in title, abstract, or keywords. • Papers that address RQ1, RQ2, or RQ3.	• Articles published prior to 2015. • Articles not written in English. • Grey literature, e.g., technical reports or dissertations. • Articles not relevant to at least 2 keywords.

Table 3.3: Inclusion and Exclusion Criteria

Accepted papers were fully read and quality assessment was checked with the following questions:

1. Is the article related to Context Awareness in S-PSS or PSS?
2. Does it propose a framework, tool or methodology?
3. Is it related to Service Design in S-PSS or PSS?
4. Is it related to UX, User-centric Design in S-PSS or PSS?
5. Is it related to industry (Industry 4.0)?
6. Do the researchers discuss any problems (limitations, threats) regarding the validity (reliability) of their results?

7. Is there a clear statement (definition) of the aims (goals, purposes, problems, motivations, objectives, and questions) of the research?

The scoring procedure assigned a value of 1 for 'YES,' 0.5 for 'PARTIALLY,' and 0 for 'NO' or 'UNKNOWN' for each question. Thus, the maximum possible score for any given paper was 7, with the cut-off set at 4.5.

3.1.4 Bibliometric analysis

This section consolidates the data from the bibliographic records of the 74 accepted papers. Analysis of the selected items can be summarised as follows:

1. *General distribution of the publications:* In total, the literature review includes 74 papers, which consists of 45 articles and 29 conference papers. Figure 3.2 presents the distribution of article types, divided into theoretical and empirical research. Theoretical research focuses on abstract ideas, concepts, and theories built on literature reviews (Marczyk et al., 2010). Theoretical articles account for 14 papers (around 21%). Six articles were classified as review papers, and include SLR papers (2 articles) and traditional review papers (4 articles). Empirical research (74% of the articles) uses scientific data or case studies for exploratory, descriptive, explanatory, or measurable findings. The contributions of these works were divided into five categories, in which the largest number corresponds to development or design approaches (23 articles), followed by frameworks (17 articles). Development or design approaches delineate the steps involved in creating a specific prototype or software, while frameworks offer a general and flexible structure for a system. The distinction lies in the specificity of the former versus the broader applicability of the latter.

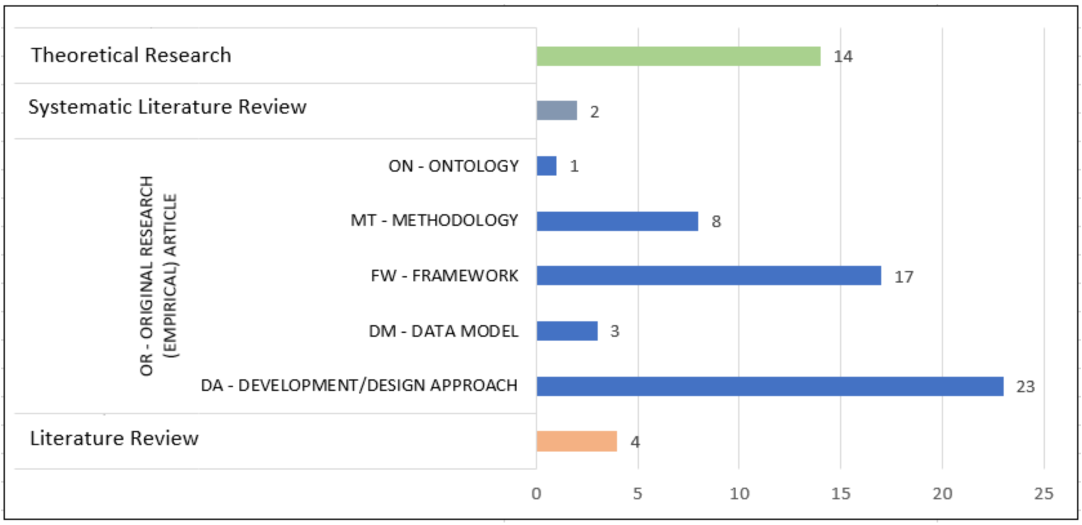


Figure 3.2: Paper distribution by contributions

Figure 3.3 and Figure 3.4 present the journal and conference distribution in descending order. *Advanced Engineering Informatics* and *Computers and Industrial Engineering* have the highest number of contributions for journals, whereas *Procedia CIRP* and *International Conference on Human-Computer Interaction, HCII* rank first for conferences.

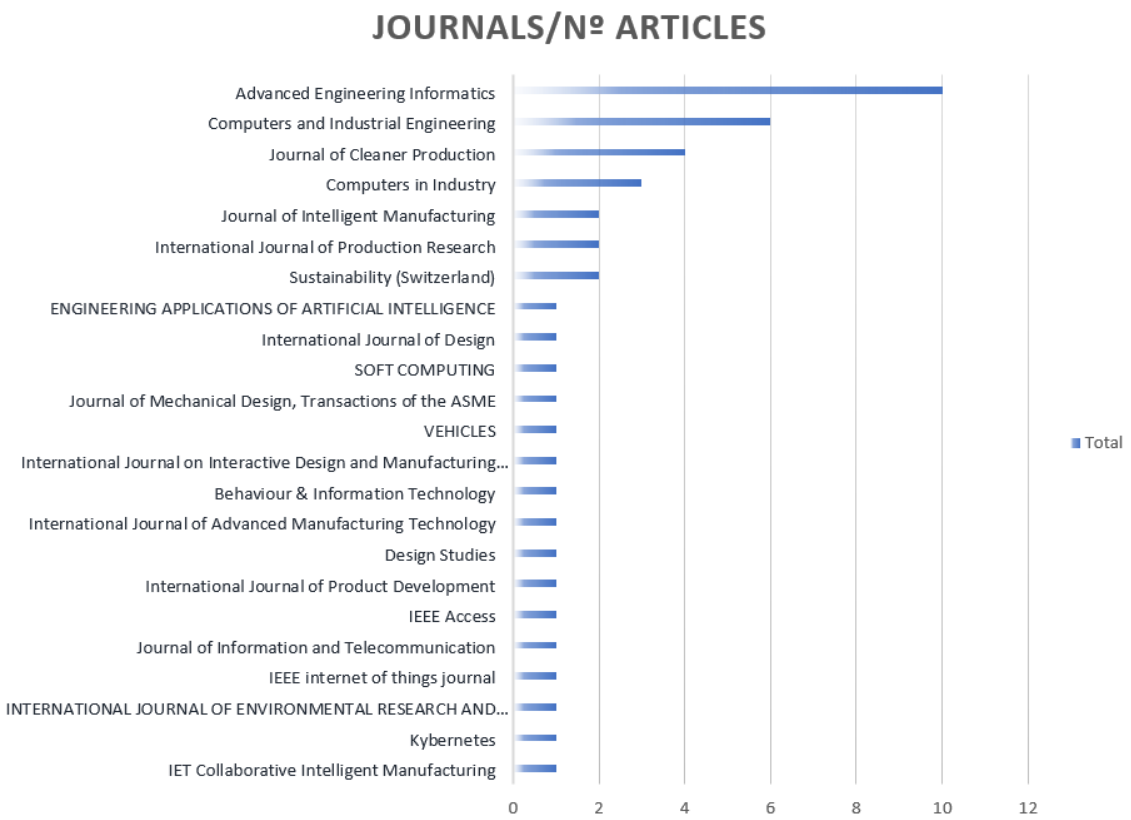


Figure 3.3: Paper distribution by Journals

2. *Case Study Distribution:* Thirty-one case studies were found in the empirical research (A.1), 15 of which followed a User-centred design (UCD) approach. Figure 3.5 shows that these studies are distributed between four industrial sectors: development of e-health and welfare S-PSS(53%), smart home applications and devices (33%), vehicles with digital services (7%), and manufacturing (related to S-PSS used on the shopfloor or devices used in manufacturing (i.e. 3D printers)(7%)). The findings are discussed in Section 3.2.

   The breakdown of the 20 case studies related to "context-awareness" is presented in Figure 3.6. These studies do not necessarily follow a UCD approach. A similar sectorial distribution can be observed: manufacturing; e-health and welfare; smart city; educational devices, and services related to products. Section 3.3 describes the more representative case studies .

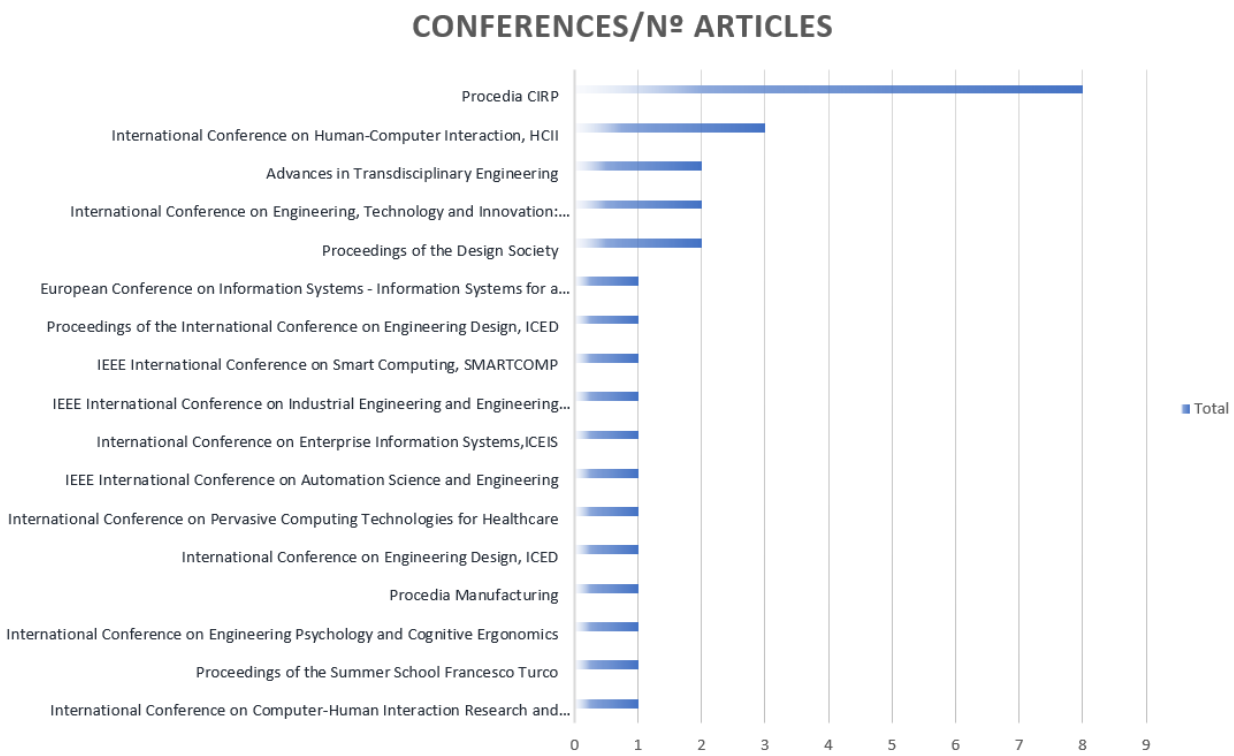


Figure 3.4: Paper distribution by Conferences

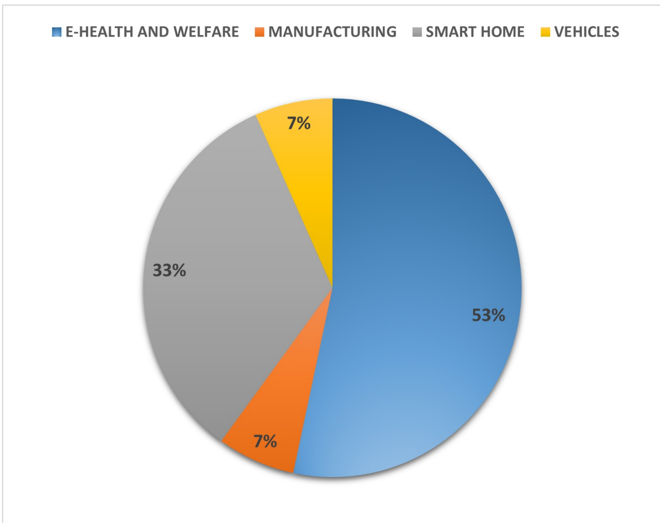


Figure 3.5: Industrial Application Sector for UCD case studies

3. *Paper distribution by year:* Figure 3.7 plots the distribution of papers per year. The final paper collection took place in January 2024. This figure shows a steady rise until 2021. The last two years show a slight decrease; however, the interest in S-PSS has remained constant. The figures in A.1 report the industrial application sector of case studies by year (Figure A.1a). The *E-health and welfare* sector presents an increase in the early years and consistency from 2020 and 2021 in case studies in this area. Similarly in context-awareness

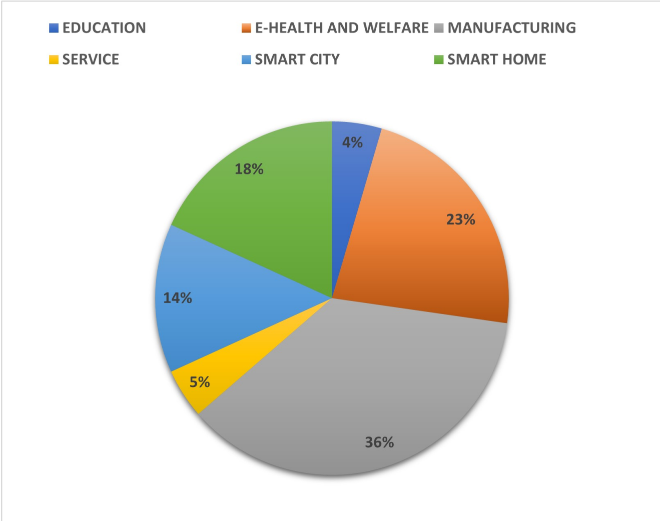


Figure 3.6: Industrial Application Sector in Context-Awareness case studies

applications (Figure A.1c), followed by the *manufacturing* sector. Especially case studies following a UCD approach (Figure A.1b), demonstrate a major increase in the *E-health and welfare* sector in 2021, which indicates the relevance of understanding user needs for end-consumers of these S-PSS types (B2C).

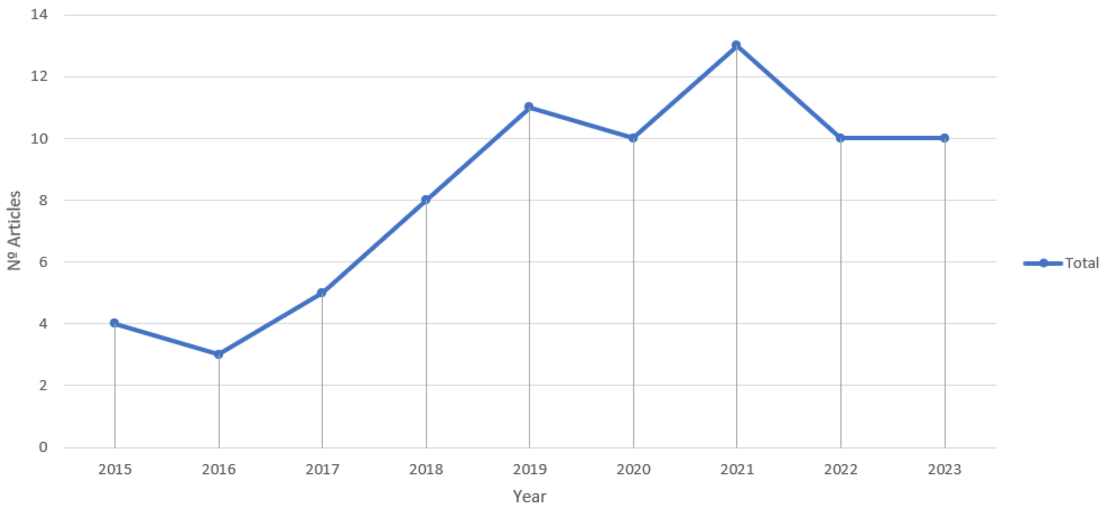


Figure 3.7: Papers distribution by years

4. *Keywords analysis:* The indexed and author keywords were extracted, and the text was normalized (removing white spaces and symbols and converting to lowercase) using the Python programming language. A cluster map was then created from the author and indexed keywords that were repeated three times or more, using the VOSViewer software (Figure 3.8), (Van Eck, Waltman, 2010) which can develop cluster maps based on bibliographic data. In the figure, node size is representative of the number of papers related to the keyword, and lines represent the links between keyword terms that have a

minimum strength of one (link with another term). The first recognizable cluster (blue) corresponds to terms like *product-service system*, *product design*, and *service design*. Of these, product design has received the most attention. The second cluster (red), refers to smart product service systems, where keywords such as *user experience* and *user-centred design* are encountered, as well as characteristics including *closed loop design* or *advance information*. The third cluster (green) corresponds to context awareness, which is related to terms like *artificial intelligence* and *cyber physical systems,* but also to *sustainability* and *sustainable development*, which indicate potential applications. In this cluster, there is no relationship between *user experience* and *context awareness*. The fourth and final cluster represents *value co creation* (25% of keyword occurrences), which is related to terms like *requirement elicitation*, *requirement engineering*, together representing 25% of the occurrences in the cluster. This is followed by *knowledge management* (12.5%) and *data driven design* (9.375%).

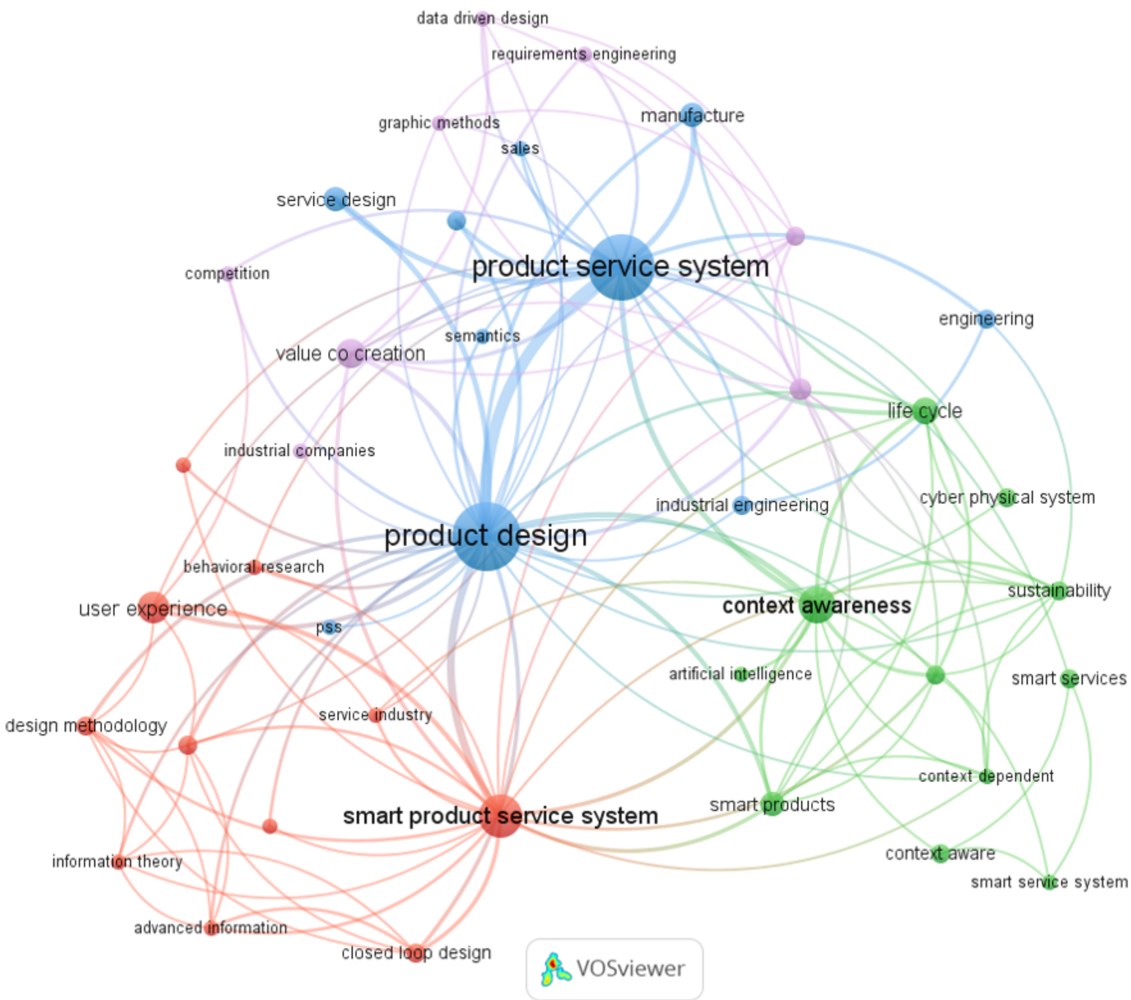


Figure 3.8: Keywords cluster map

3.2 User-centred Design in S-PSS

Based on the SLR process, this section outlines the key aspects of service design and UX in S-PSS, including definitions, research objects, and technical and application aspects.

3.2.1 Service design and data-driven value co-creation

Service design is a multidisciplinary and human-centred field whose goal is to bring new service ideas to life by better understanding of the interactions between people, institutions, and technological systems (Costa et al., 2018). Zheng et al. (2019b) state that S-PSS follows Service-Dominant (S-D) logic, a fundamental pillar of Service Design, that holds a “value co-creation” view. This perspective avocates that user experiences, context, and multi-stakeholder participation create innovative business value propositions (Wetter-Edman et al., 2014).

In the manufacturing sector, companies are attempting to add value to their products by shifting towards servitization. However, this approach may fall short as companies tend to overlook the customer experience and the views of PSS are limited to internal perspectives. In Costa et al. (2018), authors proposed an integrative approach towards PSS, based on service design and S-D logic, using qualitative research and several design artefacts across four stages: exploration, creation, testing, and planning implementation. Similarly, Liu et al. (2018) prestented a value co-creation framework for S-PSS based on the principles of S-D logic, highlighting the need for a personalized customer experience and creating value through the entire S-PSS life cycle. The framework also includes four stages: co-exist, co-design, co-implement, and co-evaluate. As Cong et al. (2020a) noted, however, data-driven value co-creation is a particular characteristic of S-PSS. The ability to capture user-generated data in real time should be an important part of the design process, and is a way of identifying and evaluating customer needs. Moreover, customers may have different roles as co-creators in the various phases of the S-PSS lifecycle: co-ideators, co-innovators, co-evaluators, co-testers, or customers as experience creators (Pezzotta et al., 2017). Each of these roles and their associated work is described in Table 3.4.

3.2.2 Closed-loop design

Closed-loop design has been described as a unique characteristic of S-PSS (Wang et al., 2019b; Cong et al., 2020a), because it is not limited to the design stage and can be extended to all phases of the lifecycle in a more holistic approach. Thus, S-PSS developers should make preparation for life-long evolution (Valencia Cardona et al., 2014). Cong et al. (2020a) defined four phases of "closed-loop design" (Figure 3.9). The opportunities for value co-creation by users and customers, here with a data-driven perspective, are identified in each phase:
(1) Requirements analysis phase: User requirements need to be identified and collected. In this phase, customers may take the role of co-ideators. In service design, studies have alluded to

Co-Creation Role	Related Work
Co-ideators: Users contribute new ideas.	(Zheng et al., 2019a; Chou, 2021; Li et al., 2020a, 2021; Zhou et al., 2022a, 2023)
Co-design: User customization of a specific product or service.	(Zheng et al., 2017; Zhou et al., 2023)
Co-innovators: Users help with the development of new concepts.	(Zheng et al., 2019a) (Hribernik et al., 2017)
Co-evaluators: Users evaluates ideas obtained through co-ideation processes.	(Wang et al., 2021a)(Mourtzis et al., 2018)
Co-testers: Users test new offerings almost ready for launch.	(Seo et al., 2016)(Bu et al., 2021)
customers as experience creators: Providers can generate richer experiences for customers using preferences and perceptions.	(Zheng et al., 2017)(Zhou et al., 2022a)

Table 3.4: Co-creation roles (Pezzotta et al., 2017)

the role of co-ideators through the automatic identification of requirements. For instance, this has been done by processing the descriptions of user experiences made in natural language (Li et al., 2020a, 2021).

It is also possible for users to participate as experience creators if behaviour and cognitive factors are included in the analysis.
(2) Innovative design phase: Using innovative prototypes, the fulfilment of requirements and customer needs can be analyzed.
(3) Design evaluation: This should consider different perspectives, sustainability aspects, value proposition, and customer value. Customers can participate as co-evaluators. Wang et al. (2020) proposed an approach for the concept evaluation of service bundles using sentiment analysis of customer opinions. An evaluation of usability and user experience is also necessary, and users take the role of co-testers, in which the use of virtual reality or augmented reality can produce immersive and more realistic experiences.
(4) Evolution phase: This phase represents the iterative and self-adaptive aspect of S-PSS, where service models or product adjustments can be triggered by the specific context (Cong et al., 2020a). In this phase, users can take the role of co-innovators. Zheng et al. (2019a) proposed a hybrid crowdsensing approach using user-generated and product-sensed information to predict design actions and incentives for users of smart-water dispensers. The role of an experience creator in a data-driven manner highlights the use of data behaviours for S-PSS customization.

3.2.3 UX in S-PSS

Functional requirements should be concurrently accompanied with cognitive and emotional requirements (Zheng et al., 2017). While service design is related to the strategic aspect of value creation, UX, as defined by ISO 9241-210, is the result of *"a person's perceptions and responses*

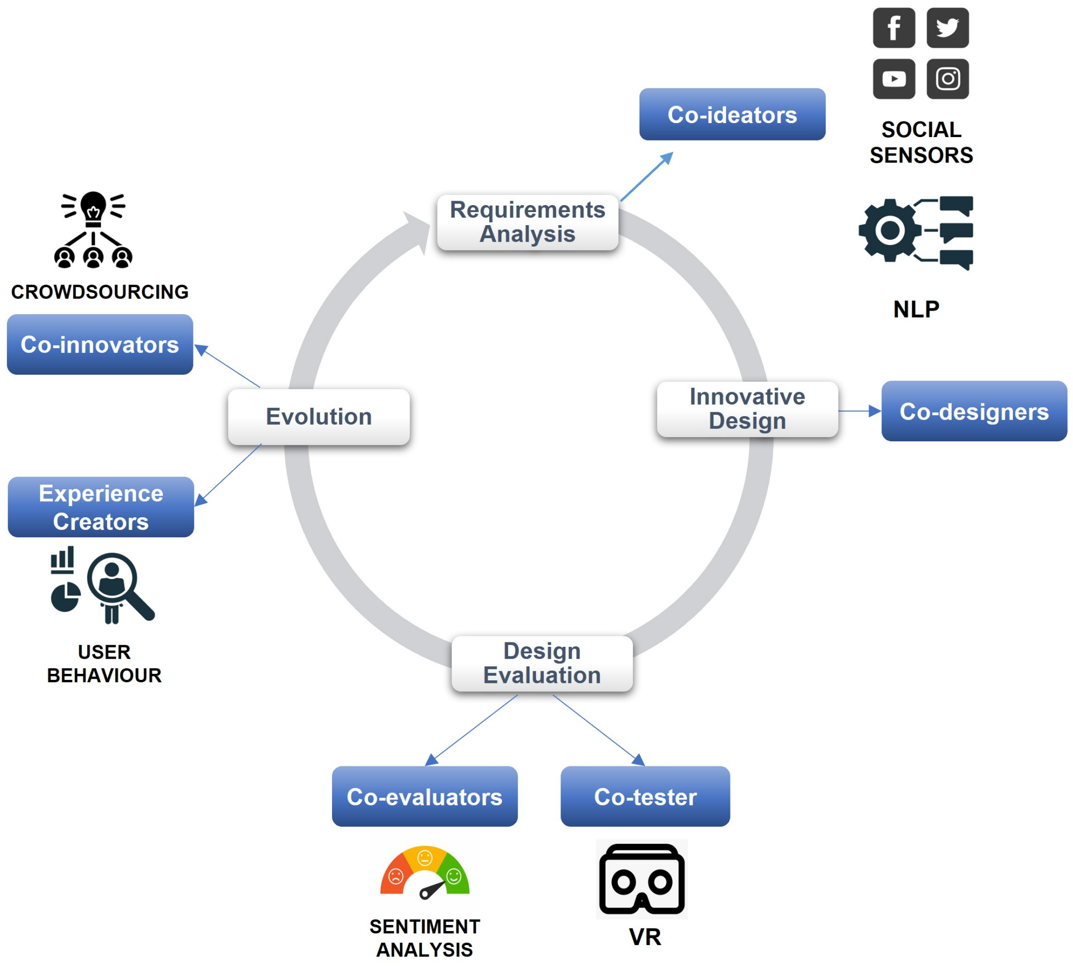


Figure 3.9: Co-creation roles and Closed-loop design

from the use and/or anticipated use of a product, system, or service" (Iso, 2010). UX emerges from the internal state of the user, the characteristics of the designed entity, and the context features where the interaction occurs (Hassenzahl, Tractinsky, 2006). The internal state of the user is represented by aspects such as predispositions, expectations, needs, and mood, which makes the user experience unique. The characteristics of the designed entity refer to purpose, usability, and functionality, where the general usability criteria from the ISO 9241-210 standard are considered a baseline (Iso, 2010). From an industrial perspective, it is necessary to take into consideration that UX for the workplace can lead to multiple UX goals. Kaasinen et al. (2015) defined UX at work as *"The way a person feels about using a product, service, or system in a work context, and how this shapes the image of oneself as a professional".*

In S-PSS, the UX is characterized by feelings of *"customer empowerment"*, *"the individualization of services"*, *"the sense of ownership"*, and an *individual and shared experience"* (Valencia et al., 2015). As experience creators, customers and users can be involved in the conceptualization of the solution, thereby supporting better customization (Pezzotta et al., 2017).

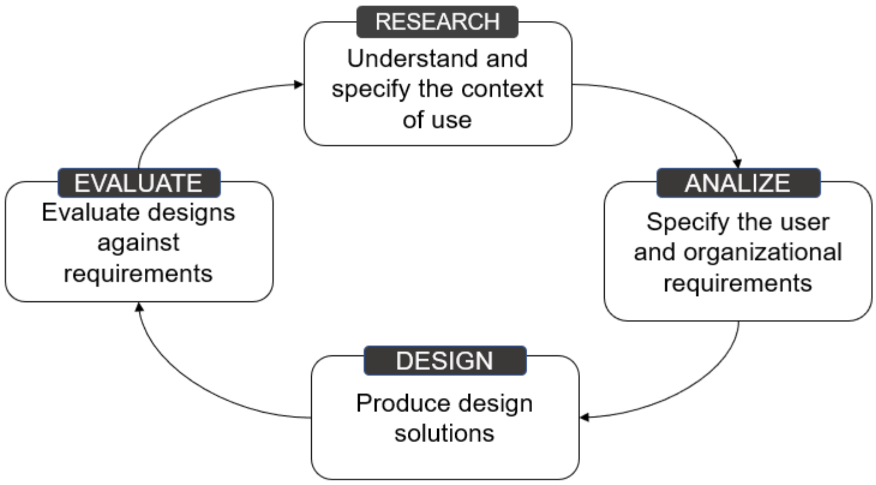


Figure 3.10: UCD process as depicted by ISO 9241-210

3.2.4 User-centred development approaches in S-PSS

User-centred development approaches place user needs and participation at the core of the co-designing development process to improve user acceptance (Ngoc et al., 2021). As a principle, UCD attempts to address and improve the entire user experience, provides a competitive advantage and contributes towards sustainability objectives (Iso, 2010). ISO 9241-210 (Iso, 2010), breaks UCD down into four stages: understand and specify the context of use, specify user requirements, produce design solutions that meet these requirements, and evaluate. The overall process is iterative and is repeated until the expected results have been obtained (see Figure 3.10).

Chang et al. (2019) presented a case study of a smart pillbox for the elderly. Using behavioural analysis and the interviews, they performed a user analysis not only to capture specific requirements, but also to understand the physiological, cognitive, and behavioural dimensions. Similarly, Jia et al. (2021) used behavioural analysis and experience maps for the development of a smart rehabilitation assistance device. In both cases, existing user problems—referred to as 'pain points'—highlighted service opportunities, and cloud-based services were developed accordingly. Non-engineering approaches and qualitative tools (i.e., observation, interviews, experience maps, etc.) are broadly used by UX designers at the very early stage of design to understand the context of use and users involved.

Validation of the design and user experience is a crucial step in which surveys and observations are an easy way to obtain feedback to measure usability and UX. Jia et al. (2021) evaluated usability based on three aspects: functionality, perceived usability, and appearance. However, questionnaires, interviews and observation can also be subjective or limited to the perceptions or personalities of participants. Methods to mitigate subjectivity include eye-tracking which is widely used by designers because of its ease of use; physiological measurement data, such as Electrocardiogram (ECG), Electroencephalography (EEG) or Functional Near Infrared

Spectroscopy (fNIRS) which help designers understand the cognitive process; and Virtual Reality (VR) which can provide a more realistic environment and reduce the physical constraints for understanding user behaviour (Dou, Qin, 2017). A different approach is the one presented by Zhou et al. (2022b) to solve concept evaluation problems in SPSS design. Using a multi-criteria decision-making algorithm to rank potential SPSS design concepts, reducing design modifications, speeding time-to-market, and improving customer acceptance.

By way of example, Bu et al. (2021) developed a VR S-PSS for a rowing machine. Using VR and fNIRS, the authors captured the information generated by the devices and the brain signals of the participants, which were then evaluated against a metric specifically chosen for the evaluated task. Seo et al. (2016) proposed a hybrid evaluation method for smart home services based on VR and Augmented Reality (AR) prototypes. Both prototypes were evaluated for possible consumers and were found to deliver similar efficiency, ease of use, and perceived usefulness. In the study, participants stated that AR provided a more natural experience of the smart home, but VR was more immersive, despite being more physically and mentally stressful. Dou, Qin (2017) obtained physiological data from several sensors, including ECG and EEG from elderly people combined with environmental information, to build a user mental model to better understand the experience of a smart TV S-PSS. User mental models also help designers to build new products and services.
Dong et al. (2019) noted that existing research is mostly focused on evaluating the UX, and thus proposed an information model that captures and represents the elements intervening in the UX of S-PSS. These included users, scenarios that are the sources of context, tangible products, intangible services, and the sequence of interactive actions. The model was tested using a graph database that helped designers to organize information more systematically. Similarly, context-based activity modelling serves as another tool for modelling the activity, the intervening actor, and the context where services are used or needs occur(Lim et al., 2017; Kim, Hong, 2019; Kim, 2023).

Table 3.5 summarizes user-centred approaches for S-PSS, including methods, tools, and needs covered.

3.3 Context-awareness in S-PSS

"Context-aware" applications have been used in the different phases of the S-PSS life cycle (context awareness is described in Section 2.3). Service providers are expected to create applications capable of reacting in varying usage scenarios. However, there exist challenges; the data is heterogeneous, multi-sourced and with different data formats, data can be derived from user behaviour (actions and values collected using the products or IT platforms) (Wutzler et al., 2017). For example, sensor data is collected directly from products (Zheng et al., 2019a), but the rise in social media awareness from individuals and organizations means it is possible to also capture the opinions and comments of customers as user-generated content.

Needs	Methods	User-Centric Design Phases			
		Research	Analyze	Design	Evaluate
Realistic Environment. Reduce physical constraints.	Virtual Reality (Seo et al., 2016; Bu et al., 2021; Dong, Liu, 2018) Augmented Reality (Seo et al., 2016)			X	X
Ease of Use.	Lo-fi prototype (Jia et al., 2021; Chang et al., 2019; Dong, Liu, 2018) Hi-fi prototype (Jia et al., 2021; Chang et al., 2019; Li et al., 2020a)			X	X
Quantitative Analysis.	Surveys (Chang et al., 2019; Jia et al., 2021; Zheng et al., 2017)	X	X		X
Qualitative Analysis.	Interviews (Chang et al., 2019; Jia et al., 2021; Li et al., 2020a) Observation (Chang et al., 2019; Jia et al., 2021)	X	X		X
Understand behaviour.	User Mental Model (Dou, Qin, 2017) User Journey Map (Chang et al., 2019; Jia et al., 2021; Lim et al., 2017; Zhou et al., 2022a)	X	X		X
Cognitive Analysis.	Brain Activity - functional Near-Infrared Spectroscopy (fNIRS), Electroencephalography (EEG) (Bu et al., 2021) Eye tracking (Dou, Qin, 2017; Zheng et al., 2017)	X	X		X
Physiological Analysis.	Electroencephalography (EEG) Electrocardiogram (ECG) (Dou, Qin, 2017)	X	X		X

Table 3.5: Methods related to design needs in UCD approaches

A common approach to measuring the concept evaluation of PSS is through key performance indicators (KPIs) from different perspectives such as customer, sustainability, or risks. Mourtzis et al. (2018) presented a context-aware framework to evaluate a PSS based on lean methodology. The sources of information were the feedback received from shop floor experts and business customers through social media platforms. They also considered the information from machines and manufacturing execution systems to obtain the KPI values. The comments and feedback were analyzed with Natural Language Processing (NLP) techniques, and the resulting information was delivered to users through a web application. In a further study, Wang et al. (2020) proposed evaluating S-PSS based on the user experience and usage scenarios, using as case study a 3D printing service. Aligned with the life cycle of context-aware applications, they modelled context sources, features, and types using key/value pairs. A set of scenarios was identified with context features and predetermined values, for instance, “printing speed”, “print frequency”, and so forth. Each scenario represented a Product Service Bundle (PSB), which was evaluated by carrying out sentiment analysis on comments relevant to the user experience. Several perceptual indicators were analyzed and ranked to identify the best performing PSB

(Wang et al., 2021a).

Another characteristic of S-PSS is their constant evolution. To fully capture the implicit requirements that users have, some authors consider it necessary to take advantage of the large amounts of data generated in the usage phase (Zheng et al., 2019b). For instance, Wang et al. (2019a, 2021b) proposed a graph-based context-aware framework for the requirement elicitation process in a data-driven manner. The framework consists of three layers: the bottom layer includes the identification of physical resources and data resources. The knowledge management layer uses domain ontologies to model the components, services, and context of a product. The top layer is the requirement elicitation layer, which is modelled through a requirements graph. Using a boilerplate presented by Arora et al. (2014), a requirement can be represented as 'under what context, system component(s) shall/should/will do process'. Each node in the graph represents the context, product, and service. The graph-based process aims to predict the most relevant product and services based on the usage context obtained through analysis of user comments. The similarity of the context node to the product and service node is ranked. Elements are listed from high to low to easily identify which requirements are more relevant to users. Similarly, it was also possible to create configurable S-PSS using a hypergraph framework and usage scenarios (Wang et al., 2021c).

In the usage phase, Ren et al. (2021) presented an approach for fault-diagnosis in smart production machines, which considers the needs of service providers in use-oriented PSS. The authors highlighted the need for big data analytics to fully potentiate the S-PSS, considering the amount of data generated for multiple customers and different products and operations. In the study, data from smart machines was modelled into tuples to form data cubes and further analyzed using Deep Neural Networks. Athanasopoulou et al. (2020) also used neural networks for path planning in a smart mobility platform which sourced context data from cameras and geolocalization. The project business model is a pay-per-service unit in result-oriented PSS.

In a paper by (Cong et al., 2022a), the author proposes leveraging real-time data during the usage stage to calculate the satisfaction degree of individual users employing machine learning techniques. Information related to the "time on task" and "task completion", plus others specific to the SCP. This approach considers various contexts of use, aligning with the closed-loop design characteristic presented in S-PSS.

Maleki et al. (2018b) developed an ontology-based framework that can support smart services in industrial machinery PSS. The study noted that the highly customized environment of industrial S-PSS could be addressed with a generic design composed of a collection of 'solution-ready' 'mix and match' components that can later be used in several PSS. Each service can be modelled into a pattern that relates the product, service, and required information, by using the modular structure of SSN ontology (Compton et al., 2011) and integrating it with context-specific classes and relationships. The framework was developed with the Apache Jena framework which is used for building semantic web applications.

Li et al. (2020b) proposed a context-aware framework for S-PSS development, that considers all phases of the S-PSS lifecycle to deliver a sustainable solution. The framework has four main steps to support decision making and optimization of the system: requirement elicitation, solution recommendation, solution evaluation, and knowledge development. At its core, context-awareness aims to model multiple scenarios in the immense amount of user-generated data and product- sensed data in the S-PSS (Li et al., 2021).

Table 3.6 summarizes each stage of the context awareness life-cycle applied to S-PSS case studies found in the literature. The research questions established in section 3.1.1 are analyzed and answered in the following section.

Case Study	Acquisition	Modelling	Reasoning	Dissemination	Lifecycle Stage
(Wang et al., 2021a) 3D Printer use-oriented B2C	3Dhub.com user-comments and Experts	Key/Value pairs	Supervised Machine Learning (SVM) NLP	Not specified	Smart Design-Usage Concept Validation (co-evaluators)
(Mourtzis et al., 2018) Mold-die making industry product-oriented B2B	Comments on issues and maintenance and manufacturing data from sensors	Relational Database	Supervised Machine Learning NLP	Web Application	Smart Design-Usage Concept Validation (co-evaluators)
(Wang et al., 2019a, 2021b) Framework evaluation using open data to simulate a use-oriented bicycle PSS B2C	Reviews and comments and sensor data	Domain Ontology, Context Ontology Graph-database (Neo4j)	Graph Algorithms	Not specified	Smart Design-Usage Requirement elicitation (co-ideators)
(Li et al., 2020b) Illustrative example of 3D Printer use-oriented PSS B2C	3Dhub.com user-comments Sensed data from 3D printer	Key/ Value Pairs, Domain Ontology	Supervised Machine Learning (Random Forest)	Not specified	Smart Design-Usage Smart End of Life
(Ren et al., 2021) Smart Production Machines use-oriented PSS B2B	Data collected by sensors. Records from MES.	Data Warehouse	Deep Neural Networks	Not specified	Smart Maintenance Fault Prognosis
(Athanasopoulou et al., 2020) Smart Mobility Platform result-oriented PSS B2C	Camera data. Sensor location information.	Not specified	Neural Networks	Not specified	Smart Usage

Case Study	**Acquisition**	**Modelling**	**Reasoning**	**Dissemination**	**Lifecycle Stage**
(Maleki et al., 2018b) Machine health monitoring product-oriented PSS B2B	Sensors in laser cutting machines.	PSS Ontology and SSN Ontology	Rule Based reasoning and semantic reasoning	web application	Smart Maintenance
(Seo et al., 2016) Smart home product-oriented PSS B2C	VR devices and AR app interactions	UX ontology for smart home services	Semantic reasoning (Jean inference)	mobile app	Smart Design Smart prototyping (co-testers)
(Le et al., 2020) Smart bot for Customer Support product-oriented PSS B2C	Forums related to product (extracting questions and answers)	Context-aware knowledge Ontology	NLP based on RASA NLU Open Source	web application	Smart Usage
(Cong et al., 2022a) Surgical robot for flexible ureteroscopy product-oriented PSS B2C	Data collected in the usage context	-	BP Neural Networks	-	Smart Design and Evaluation
(Yuan et al., 2023) product-oriented PSS B2C	Physiological data acquired from users through wearable devices	Entity Relational Model (Mysql)	Neural networks for context detection model and rule-based for service recommendations.	Monitoring center interface	Smart Usage

Table 3.6: Context aware case studies found in the literature

3.3.1 Recommendation Systems and Context-awareness

Context-Aware Recommendation Systems (CARS) have also been identified in the literature. In general, Recommendation Systems (RS) are intelligent algorithms that analyze user preferences and behavior to provide personalized suggestions or recommendations for products, services, and content (Jannach et al., 2010). The goals and uses of RS for S-PSS can be:

- Enhance the user experience: Providing personalized recommendations for services based on user preferences, history, and context.
- Increase adoption of services: Promoting less-known services to users that are relevant in a given context, so as to facilitate their adoption and usage.
- Improve customer retention: Providing personalized recommendations to users that maintain engagement with the PSS application.
- Maximize the value of the PSS: Suggesting the most relevant and valuable service configuration to the user, thereby increasing the overall value of the PSS for the user (Zhang et al., 2023; Chiu et al., 2021).
- Provide 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. As stated in Chapter 2, a context-aware system uses context to deliver information and/or services relevant to the tasks of a user (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).

The presented SLR has shown that a substantial amount of the work to date in context-aware applications for S-PSS has been dedicated to concept evaluation of product-service bundles, and requirement elicitation and analysis for capturing implicit requirements. For these purposes, the use of external sources for smart products, such as social sensors, has increased dramatically. 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., 2021a; Le et al., 2020; Li et al., 2020a; Lin et al., 2016). However context can be also used to resolve the services or information which need to be presented to the user. Hence, 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 the ability of a system to accommodate the physical and mental abilities of individual users, as well as the context of use and platform capabilities (Stephanidis, 2001).

Case Study	**Objective**	**Data used**	**Technique**	**Lifecycle Stage**
(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.
(Esheiba et al., 2021) Laser machine for turbine engine manufacturer	Hybrid knowledge-based recommendation 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., 2022b) 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., 2022b) 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	Conceptual framework for an SPSS 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).
(Ren et al., 2023) Smart reading service system	A method that includes context awareness, interaction solution recommendation, and the establishment of an Interaction Design Management Guide (IDMG) to achieve proactive and personalized interaction solutions.	Eye movement data. Stress State Self-report Questionnaire (SSSQ).	Decision-Making Graph and Euclidean distance for similarity measure.	Smart user experience.

Table 3.7: Context-Aware Recommendation Systems (CARS) case studies found in the literature relevant to S-PSS

According to Adomavicius, Tuzhilin (2010), the growing availability of contextual information and the need for personalized and relevant recommendations have led to the rising popularity of CARS. By leveraging contextual information, these systems can provide more accurate and useful recommendations for 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 in which contextual data is processed (Papadakis et al., 2022). The first is *contextual pre-filtering*, in which contextual data is 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 context. Lastly, the *contextual modeling* approach involves the direct integration of contextual information into the recommendation model, such as incorporation into the preference computation process.

In the literature, RS have been used in S-PSS for a number of purposes. Esheiba et al. (2021) proposed a hybrid knowledge-based RS that recommends the customer a set of product customization requirements and design parameters 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 proves challenging once the smart product is in use, as it requires active user participation or scrapings from external sites.

Table 3.7 summarizes published CARS case studies relevant to S-PSS, detailing objectives, data sources, applied techniques, and the specific stages in the S-PSS lifecycle to which the case study or approach was applied. From the table it is observed that these studies (Esheiba et al., 2021; Cong et al., 2022b,a) focused on the recommendation of an optimal design configuration for customers based on previous configurations. Focusing on a smart usage, the work presented by (Chiu et al., 2021) is a development approach that is very focused to the tourism industry, the data used comes from external sources, in this case Google maps API, there is not a methodically description of how adaptations would form on the UI. Recent works from Yuan et al. (2023) and Ren et al. (2023) exploit data sources internal to the S-PSS. With the goal to deliver an smart user experience and usage. Yuan et al. (2023) presents a development approach that employs data from smart wearables and user profiles, the contribution of this work is the analysis of primary contextual data to produce generalized contextual information that can be later used to produce service recommendations.

3.4 Discussion and Opportunities for Research

To answer the research questions carefully and comprehensively, our analysis considered significant aspects related to context awareness capability in S-PSS, with a particular focus on design and the user experience.

RQ1: What relationships exist between UCD, UX, and Context-Awareness in the context of S-PSS

Context is any type of information which assists in identifying the existing situation of an entity, be it a person, place, or object. Several researchers have highlighted the importance of context in S-PSS (Cong et al., 2020a; Valencia Cardona et al., 2014; Zheng et al., 2019b) because it helps ensure that services and products can adapt in multiple scenarios and various stakeholders. UCD and its relationship with service design and UX is explored in greater detail in Section 3.2. Figure 3.11 shows how these concepts, which are interrelated in the design of S-PSS, use *context*. S-D logic as part of service design focuses more on the creation of value

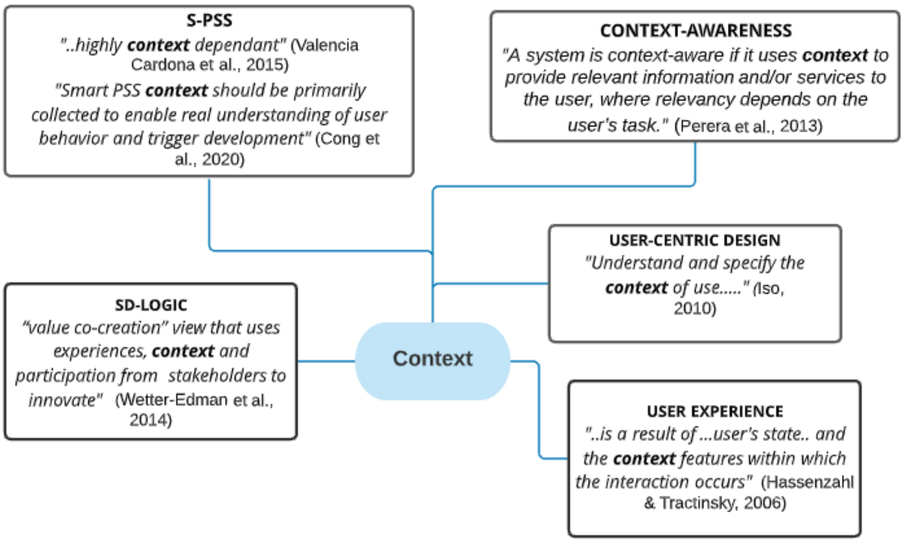


Figure 3.11: Context relationship in an S-PSS ecosystem based on: (Valencia et al., 2015; Cong et al., 2020a; Perera et al., 2013; Iso, 2010; Hassenzahl, Tractinsky, 2006; Wetter-Edman et al., 2014)

for the customer. It is a co-creation process and a great part of the value is in the context, which suggests that the process must be capable of understanding the environment of the beneficiary and associated resources (Wetter-Edman et al., 2014). UX concentrates on the part of interaction with the S-PSS, whose sources of context are a combination of user environment, identity and tasks to fulfill, all of which affect the experience. Context-awareness capability is thus critical to the S-PSS, and context must be captured for customers and users.

RQ2: How has context awareness capability been used in the design of S-PSS??

Section 3.3 explored case studies found in the literature whose contribution is reflected in frameworks and development approaches. Table 3.6 shows that these case studies were not limited to previously established design stages, but rather were extended to the maintenance of smart products. To address the research question, the important aspects of the stages involved in the development of context-aware applications must be analyzed to finally highlight the uses in UCD and the user experience.

•*Acquiring and Modelling context*

The specifications of concern in S-PSS can be abstracted in a model. Data acquisition from heterogeneous sources necessitates the use of data models in all the studies reviewed, and they are also a fundamental step in context-aware applications. As regards S-PSS design, physical sensor data from smart products, and also social sensor and user data must be considered. However, there is a static component to S-PSS that represents the structure of the smart product and related services (i.e. bundles), which are usually defined before usage. According to Maleki et al.

(2018b), the domain knowledge models in an S-PSS architecture function as intermediaries in the integration of S-PSS services and cyber-physical components. One approach is the use of domain ontologies. Ontologies are a means to formally model concepts from a particular domain into a detailed specification of entities with properties and relationships. Domain ontologies define the vocabulary related to a particular domain (Gruber, 1993; Guarino et al., 2009). Some authors have developed new ontologies (Le Dinh et al., 2021; Wang et al., 2019a). Hajimohammadi et al. (2017) proposed a generic ontology for PSS development that includes all the life cycle phases of products and services. In contrast, others have adapted existing ontologies and integrated these into their solutions. Li et al. (2020a) used the Function-Behaviour-Structure (FBS) ontology to model the prototype of a smart product. User-generated data collected from Wiki pages and e-commerce platforms was exploited as a source of context to create a knowledge graph using Neo4j. The resulting solution could provide innovative concepts and probable relations when solving unusual requirements for S-PSS.

The context acquired from social sensors can be derived from multiple sources (i.e., social networks, emails, e-commerce reviews). In these cases, the modeling and reasoning process facilitates the determination of context, such as usage scenarios, user types, places, or times.

•*Reasoning context*

Most of the studies presented in Table 3.6 used supervised machine learning techniques and NLP to address the reasoning phase. NLP is a subfield of machine learning and is mainly used in raw and unstructured text. Machine learning takes advantage of the large amount of data generated by smart devices, social sensors, and users. However, some of these studies also point to a lack of data as a challenge in the early phases of S-PSS (Wang et al., 2020). In such cases, the selection of hybrid methods that combine one or more techniques can be employed. If ontologies are used in the modelling stage, semantic reasoning could leverage this to infer new knowledge based on the existing ontological relationships using frameworks such as Jena (an open source Java framework for building semantic web applications). Graph databases can also import ontologies, generating graph algorithms to infer new information. For instance, Neo4j—a graph data platform that includes a database and data analysis suite—has been used in some of the works presented in this review (Seo et al., 2016; Maleki et al., 2018b).

•*Dissemination and Monitoring*

The dissemination of 'context' to end-users, its application in the *usage* stage of S-PSS, and the target users of the solutions was not generally addressed in the reviewed papers. Nonetheless, dissemination is particularly relevant to user experience and user interaction, since it is necessary to consider how new information or services are delivered to the user. Hence, this aspect could require input from more perspectives, for example, designers, service providers, and end-users.

Furthermore, the process of context monitoring represents the evolution of the design. Feedback about decisions made by the application from integrated surveys and user behaviour should be monitored. For instance, if a user is not happy with the service or information offered by the application, it is likely they will return to an earlier version of the application, or decline to interact with the information, both of which can be construed as negative feedback. In such scenarios, reinforcement learning can help to maximize some notion of reward, for instance, end-user satisfaction by optimizing the recommendations, information, or services provided to the user (Nurmi, Floréen, 2004). *No evidence of methods to address the evolution of a context-aware applications used in S-PSS was found in the literature, however.*

•Context-Awareness in User-centred design (UCD)

In the domain of S-PSS design, a significant part of the utilization of context-aware capability has been directed towards evaluating product-service bundles, and eliciting and analyzing requirements to capture implicit needs.

These approaches often make use of external sources to the smart products such as social sensors. Many case studies coincide in the analysis of user comments related to user experiences, obtained through channels like social networks. Techniques like NLP techniques, and sentiment analysis are commonly employed to analyze and quantify sentiments expressed in these comments.

Hence, new requirements emerge from the feedback of product-service bundles. In such scenarios, context-aware applications take a passive role; they present new or updated information to an interested user or they make the context persistent for the user to retrieve later (Liu et al., 2011). Overall, the role of users as co-ideators has been largely explored in the literature (Zheng et al., 2019a; Chou, 2021; Li et al., 2020a, 2021; Zhou et al., 2022a, 2023); requirement analysis and concept evaluations are tools for internal stakeholders to make decisions. *However, there is no evidence in the literature of an evaluation and the use of the results by companies; a more industrialized view is necessary.* There is still room in other areas to present more studies, for instance, in the areas of adaptive design and evolution.

In these areas, *there is a need to further integrate real-time sensor data from smart products and leverage it as a source of context. The utilization of this contextual data should provide adaptive applications tailored to each use case. Furthermore, a research gap exists in using data from user behaviour and preferences during the usage stage of S-PSS from internal sources.* This gap underscores the need for the development of active context-awareness applications that automatically adapt to context through alterations in the behaviour of the application or presentation of services (Liu et al., 2011).

As for the user experience, context-awareness has been used to create more realistic prototypes that can adapt to usage, hence obtaining a realistic opinion of the product before fabrication (Seo et al., 2016). Another use of context-awareness is the creation of mental models from users as they test smart products, by capturing data from a variety of sensors devices specifically designed to obtain physiological and psychological information (Dou, Qin, 2017).

Thus it can be seen that the context-aware applications presented in the literature attempt to capture or perceive the context to bring new information, but no methods to adapt the S-PSS to the context are proposed.

•*Current Challenges*

For context-aware applications, there is little published data which focuses on architectural patterns which support the modularity, interoperability, and scalability of systems in S-PSS scenarios whose operations foresee a large number of simultaneous users. Thus, to be able to respond to the real-time needs of multiple users, the volume of data, the rate at which data is produced, and the variety of sources must be considered in big data analysis and architecture. The process of filtering and validating the relevant contextual information within the time limits specified by the each application is a difficult process. Consequently, research on real-time and context-aware big data processing techniques for managing contextual information of any business entity instantly, including internal and external sources of context, is warranted (Dinh et al., 2020).

RQ3: What are the current gaps and challenges in the design of S-PSS to satisfy user needs and improve UX?

Most engineering methodologies for design are oriented towards smart products rather than smart services (Cong et al., 2020a). It is therefore necessary to develop more frameworks or methodologies to support the *smart service* aspect of S-PSS. Other important aspects to consider are the following:

•*Adaptive design in S-PSS*

The 'Closed-loop design' characteristic highlights the constant evolution of an S-PSS, and although requirement management and elicitation have been analyzed, little attention has been paid to strategies to manage user feedback and behaviour (Pirola et al., 2020). *Studies on UX in S-PSS are particularly scarce, and those that do exist focus primarily on the evaluation of user experience in the design stage.* Thus, they do not follow the holistic perspective of the S-PSS life cycle. Considering the design characteristics established in section 3.2, most of the reviewed works do not address the exploitation of data generated from devices and users to provide 'customized experiences' and 'self-adaptable design' which can react to multiple sources of context (i.e. device context, user context, and environment context). Cong et al. (2020a) stated that user preferences should be associated with different design elements of Smart PSS in specific usage contexts. However, the published data is still limited to some user-specific preferences in the usage stage of Smart PSS. For example, digital service platforms accessible through mobile apps or web applications play a significant role in the S-PSS offering. The use of Adaptive User Interface (AUI) in this context has the potential to adapt to user interaction patterns, and hence achieve a more personalised UX that can lead to greater user satisfaction in S-PSS . *We posit that the role of users as "experience creators" and "co-designers" can be further exploited to provide S-PSS customization using a data-driven approach.*

•Data Privacy

Data-driven design will require user personal and behavioural information, which could vary depending on S-PSS application domains. The services created must help users make informed decisions about their privacy. Similarly, users must have control over their personal information (Ooijen van, Vrabec, 2019). Privacy regulations should be followed (i.e. General Data Privacy Regulation GDPR) and ensure users' consent and awareness of the data collected (Zheng et al., 2019a). Furthermore, considering the use of several machine learning approaches reviewed in this chapter, it is necessary to define "how much data", "what data" and "why" .

•Holistic UCD view and Multi-business perspective

UCD is crucial for capturing the needs of stakeholders and guaranteeing a better user experience. However, studies using this approach are limited to the *design* stage of S-PSS, and the continual evolution of smart products and services whilst maintaining a user-centric approach remains challenging. Moreover, 53% of the reviewed case studies that used the UCD approach were related to *e-health and welfare* S-PSS, followed by 33% in Smart Home applications. It seems that sectors of this type, which are characterized by a Business-to-Consumer (B2C) business model, are more interested in understanding final consumers, since they represent a large part of the market to which their products and services are directed. In contrast, far fewer case studies following a UCD approach were found in *manufacturing*, as such sectors are characterized by a Business-to Business model more focused on selling industrial products such as materials and parts, capital items, supplies, etc; an area where users are not always so evident or considered (He, Zhang, 2022).

Seven out of the ten case studies reviewed were focused on B2C (Table 3.6), and three case studies were directed towards B2B. Although basic design and UX principles are shared between both approaches (i.e. functionality, intuitiveness, attractiveness, etc.), the goals of the user experience change—end-consumers are strongly motivated by emotions, in contrast with B2B models, which respond to professionalism, customization of services, and establishing loyal relationships (Ritter, Winterbottom, 2017). For this reason, obtaining opinions from B2B S-PSS from social sensors will be difficult, and more integrative ways to gather feedback within the S-PSS will be required.

Thus it can be seen that S-PSS have attracted considerable academic interest in recent years. This review followed an SLR approach, which presented some limitations since the wide range of topics covered made it difficult to classify the selected works. However, the review does provide the reader with a general overview of concepts, and presents the current state of the art, as well as gaps and challenges. Moreover, the three research questions have a been addressed and the relationship between UCD, UX, and context-awareness in S-PSS has been synthesized.

3.5 Conclusion

As a business strategy that combines smart services and connected products into one solution, S-PSS have gained more attention among academics. The current review followed an SLR approach and presented some limitations because of the breadth of the topics covered , which made it difficult when classifying the selected works. However, the review accomplished its goal to provide the reader with a general overview of concepts while presenting current studies described in the literature, gaps and challenges. Furthermore, the current study has answered the three research questions and synthesized the relationships among UCD, UX and context-awareness in S-PSS.

Other contributions of this work are as follows:

- *A bibliometric analysis of the SLR* in the field of S-PSS design with a user-centred approach. This includes the description of both empirical and theoretical research. Furthermore, an analysis of industrial sectors in cases of studies following context-aware and UCD approaches. Finally, a keyword cluster analysis that provides a visual and quick overview of the subjects that the studies have focused on and how they are interrelated.
- *An analysis of design aspects of S-PSS with a user perspective* The present chapter examined the aspects and internal characteristics of the design of S-PSS as related to different roles that users can have in design in the various stages of the S-PSS life cycle, enhancing the digital capabilities of S-PSS that allow "data-driven value co-creation" within a "closed-loop design approach". The review has presented works following UCD, which have been utilized only in the design stage. This is true, especially in the very early stages of S-PSS development, where the lack of data can hinder a data-driven design. The authors emphasize that the personalization of services based on context could allow the design to be extended to the usage phase of S-PSS and provide a smart user experience.
- *A context-aware lifecycle analysis of case studies* Following the lifecycle of context-aware applications, this work have described each stage and the multiple case studies found in the literature have been synthesized.
- *Identification of Key Gaps and Future Research Directions:* In the literature, important gaps have been identified. There is a gap in integrating real-time data from smart products to provide adaptive applications to each use case, particularly during the usage stage. Additionally, the dissemination of the results of context-aware approaches to end-users and monitoring the evolution of context-aware applications, especially during usage, is not extensively addressed in the literature.

  UCD is mainly applied in the design stage of S-PSS, and there is a challenge in maintaining a user-centric approach throughout the continual evolution of smart products and services. This shift towards data-driven design in S-PSS may require personal and behavioral information from users. Addressing data privacy concerns and ensuring users have control over their personal information is crucial.

This chapter served as the foundation for forming hypotheses and identifying research gaps in the literature. After examining the relationships among various components of the design of S-PSS and context information, the identified requirements were the creation of new methods to provide a smart user experience in S-PSS. These methods should take into consideration a user-centric design as a basis, consider the data-driven co-creation characteristic of S-PSS from internal data sources. Furthermore, the importance of context-awareness.

4 AdaptUI: A context-aware framework for adaptive user interfaces in S-PSS

The literature review presented in Chapter 3, provides an analysis of works relevant to various aspects and stages of S-PSS design. This has revealed three essential characteristics and requirements that should be considered when building an S-PSS ecosystem.

- **Data-driven value co-creation:** The value co-creation emphasizes the role of users in the design of S-PSS. Various roles identified in the literature have been examined. However, S-PSS digital capabilities allows to employ user-data collected in real time to align design elements with user preferences (Cong et al., 2020a). In this context, a gap in the role users play as "experience creators" is highlighted.
- **Closed-loop design:** The design of services cannot be static; it must be an iterative process and include continuous improvements (Zheng et al., 2019b). This principle should be applied in all stages of the design process, encompassing the incorporation of new requirements, refining the presentation of digital services, and evaluating and monitoring any introduced changes. The requirement analysis stage has received a great part of attention in the literature, this represents opportunities in other aspects of the design.
- **Context awareness:** It is a fundamental requirement for all components that affect the ecosystem of S-PSS. In this sense, the role of context-awareness applications can be passive, perceiving contexts and generating new information that allows for the creation of new requirements. They can also contribute to active context-aware applications that dynamically adapt their behavior or provide context-driven responses (Perera et al., 2013). From the SLR, many applications use context to provide new information specially on the stages of requirement elicitation and concept validation and evaluation. However, when coming to usage of the S-PSS is necessary that designed context-aware applications are active to react to the contextual data and produce adaptations to the current behaviour of the S-PSS

From this analysis and the literature, this work addresses these critical research gaps:

- 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
- Very few studies have focused on the 'usage' stage in the S-PSS lifecycle, in which user interfaces facilitate communication between users and smart products. Real-time user interface adaptation that respond to individual user needs, remains an underexplored area with the potential to significantly enhance user experiences. Additionally, existing studies in the literature have focused more on methodologies and frameworks addressing the physical aspects of S-PSS, overlooking the smart service dimension of the system (Cong et al., 2020a).
- The absence of software architectures or frameworks for AUI development is a significant challenge, as revealed in a recent systematic mapping review 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).

In this chapter, therefore, we propose a framework for real-time user interface adaptation and present applications that dynamically adapt user interfaces to user behaviors and contextual changes within the S-PSS ecosystem. This framework aims to enhance the UX of Smart Product Service Systems while considering their inherent requirements.

4.1 Framework Description

This research project introduces AdaptUI, a framework designed as a conceptual structure to organize and guide system development. It encompasses predefined components and guidelines, offering a structured approach. The primary goal of this framework is to facilitate the generation of AUIs within the dynamic context of S-PSS. As presented on Chapter 2.4, the AUI development is supported by three pillars: AI techniques, user modeling and human-computer interaction (i.e. usability testing and UCD design)

In the framework, automation of these interfaces is driven by context-aware recommendation systems, a technological approach identified through the state of the art (Chapter 3). These systems, which tailor content based on real-time contextual factors, can be especially useful in the context of smart services, where digital platforms present an array of e-services for users. Recommendation systems and techniques such as collaborative filtering can specifically exploit user data to provide recommendations.

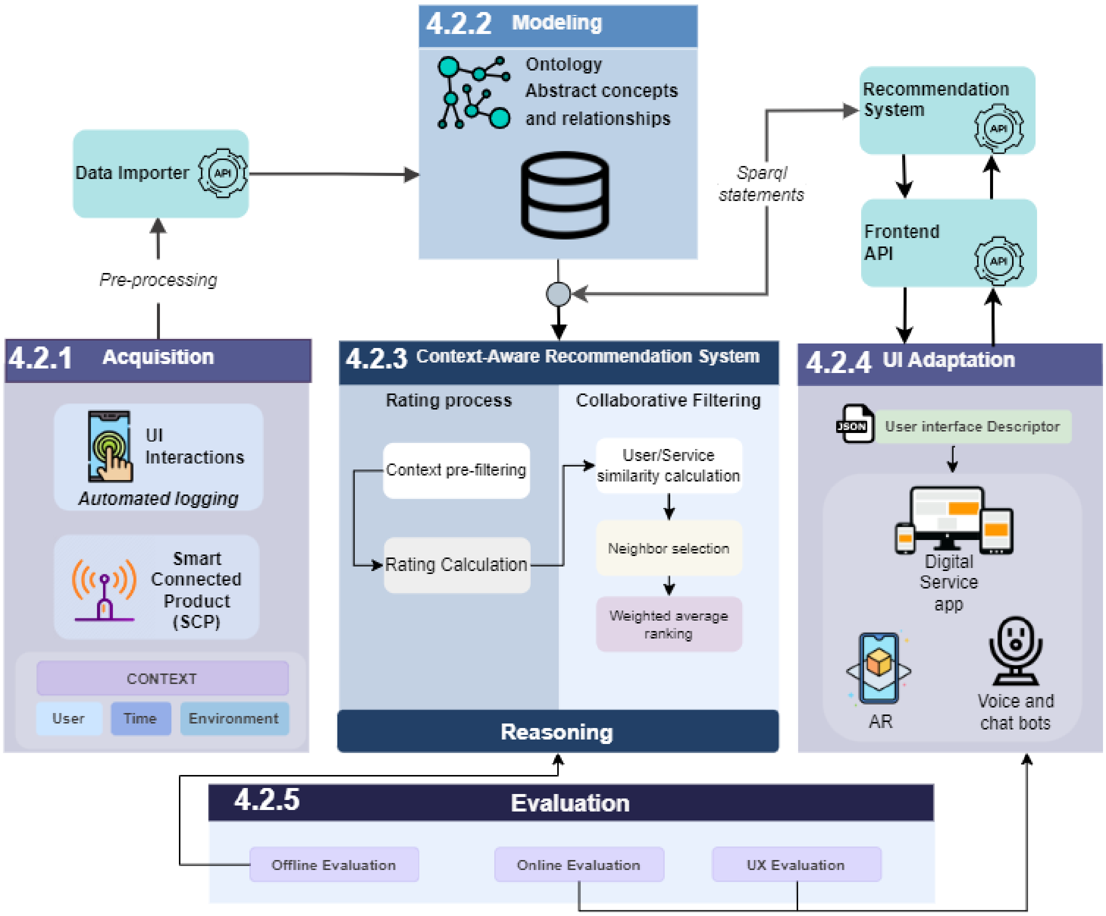


Figure 4.1: AdaptUI: Context-aware framework for adaptive user interfaces in S-PSS

The framework is depicted in Figure 4.1 and is based on the four general components of the context-life cycle (Perera et al., 2013), adapted to emphasize user interface adaptation. Each of these stages will be described in the following subsections.

4.1.1 Acquisition

Acquisition refers to the process of collecting data from Smart Connected Products (SCPs) together with data relevant to the user and the way in which they interact with products (Pulikottil et al., 2023). SCPs are equipped with sensors, embedded systems, and communication capabilities that facilitate collection of various types of data, such as sensor readings, usage patterns, and environmental information. Interaction data, on the other hand, helps infer user behaviour and serves as a way to understand users.

The process of acquiring user interactions depends on the way users interact with the machine or product, and is usually associated with the visual elements of the interfaces. In our framework, the automated logging of user interactions is proposed as the data collection method. This offers several advantages including non-intrusiveness, real-time data capture, and minimization of errors associated with manual data entry. Specialized software such as 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 into the system. However, this approach can be challenging for complex interfaces and may require access to the underlying code of the system. 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, together with timestamps. The output was then transformed into a UI log through image-analysis techniques.

Data acquired in the methods described above can provide contextual information relevant to the use of digital services offered by the product. The SCP itself is also a valuable source of contextual information that can help the adaptation process, 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 overly specific contextual information.

4.1.2 Modeling

The modeling stage provides 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 are used to formally model concepts from a specific domain by providing a detailed specification of entities, their properties, and their relationships (Guarino et al., 2009). In effect, they define the vocabulary associated with the respective domain, enabling a structured representation and organization of knowledge. The UPON methodology derived from the Unified Software Development Process (De Nicola et al., 2005) serves as a guide for ontology building. In the present framework, the goal of the ontology was defined from the outset as: "to represent digital services offered by an S-PSS and support the adaptation of user interfaces". Then competency questions (CQs) were defined to describe the type of question the ontology is expected to answer. The ability of an ontology to answer these CQs is a crucial functional requirement, and serves as a way to evaluate the ontology (Ren et al., 2014). Three CQs were defined:

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 was employed to formally define the concepts and relationships between the entities in the ontology (Figure 4.2). In the top-down approach, concepts go from general to particular. The methodology highlights the benefits of reusing existing ontologies, and thus ontologies that model the main concepts of S-PSS and contextual information specifically for

AUI were selected. 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.

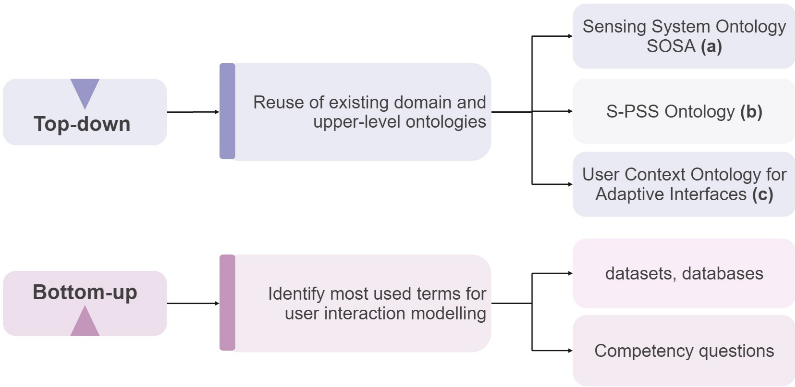


Figure 4.2: Ontology design hybrid approach. (a)Janowicz et al. (2019) (b)Maleki et al. (2018a) (c)Iqbal et al. (2021)

The resulting ontology, Service Interaction Context Ontology (SICO) [1], was created in Protégé and an overview of the general structure is presented in Figure 4.3. To visually represent the components of the ontology, we divided the entities into three groups: S-PSS, Context, and Interaction.

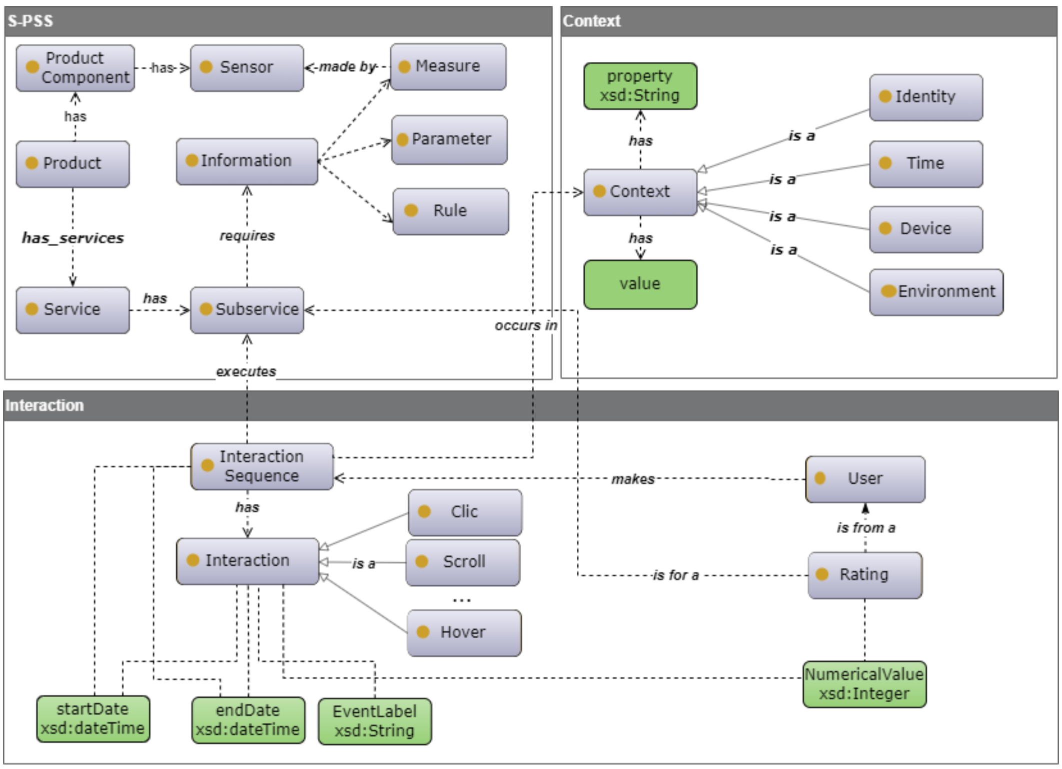


Figure 4.3: Ontology general view

[1]https://aicarrera.github.io/SICO/index-en.html

S-PSS: In an *S-PSS*, a static model is necessary to represent the services of smart devices. In accordance with Maleki et al. (2018a), each service is modeled into a pattern that relates the service with the product and required information. Services may be 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 is dynamic, it needs to be created as instances or individuals derived from these types for each particular implementation of the framework, as illustrated in the example on Figure 4.4, where light blue elements represent individuals corresponding to each particular context category, in this case Identity and Time.

Figure 4.4: Context Entities and Individuals example

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 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 $S =< e_1, e_2, ..., e_m >$ $(e_i \in S)$ is an ordered list of events e_i occurring in a user session, where S is a set of known events whose 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 is assigned a numerical value based on the actions triggered. The set of generated sequences is then linked to a specific service in the application, which ensures a comprehensive understanding of user interactions within the context of that service .

The use of ontologies in the framework delivers a flexible interrelationship of elements in a natural manner, unconstrained by specific data types. Ontologies also ensure the derivation of meaningful inferences from established relationships, a feature challenging to achieve in traditional database design.

Ontology validation

The design of the ontology presented in this work was evaluated using the OOPS! tool (Poveda-Villalón et al., 2014). This tool assesses the ontology against a set of 41 pitfalls recognized as the most common in ontology design. These pitfalls are categorized based on three levels of importance: *critical*—crucial to correct, as neglecting to do so may impact ontology consistency, reasoning, and applicability; *important*; and *minor*.

A preliminary analysis revealed that the ontology had:

- 1 critical issue (P19: Defining multiple domains or ranges in properties),
- 2 important issues (P34: Untyped class, P41: No license declared), and
- 5 minor issues (P08: Missing annotations, P04: Creating unconnected ontology elements,

P22: Using different naming conventions in the ontology, P13: Inverse relationships not explicitly declared).

This analysis helped improve the ontology and address all critical and important issues. The ontology now only has 2 minor pitfalls that do not affect its functioning. These are: P08, Missing annotations, which provide additional information about the fields and data properties. However, this information has already been provided in the ontology publication. P22, using different naming conventions in the ontology, which does not refer to any specific field and is general to the ontology. We reviewed the ontology and verified that all object properties and data properties followed the correct use of delimiters (_).

Then the ontology was validated against the requirements it must fulfill. The approach for this part of the validation was to determine if the ontology could answer the competency questions that represented those requirements (Ren et al., 2014). Each question was transformed into a SPARQL construct, which was deployed using the "visual graph visualizer" of the RDF database GraphDB [2].

- *CQ1: In which contexts is a service executed?*

  Figure 4.5 illustrates the SPARQL query which indicates the contexts in which a service is executed. It retrieves the interaction sequences that *executes* a service, linking them to the corresponding context through the property onto:occurs_in. The resulting triples connect the service to its execution contexts based on the RDF data. As observed in the context individual (context23), it belongs to a specific subclass, in this case "Identity" context, and also includes the name description of the contextual information and the value.

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

  The SPARQL query indicating which service or sub-services are utilized by users is depicted in Figure 4.6 . The query connects users with interactions in which they participate (:makes), to the services executed in those interactions (:executes). From there it can be inferred which user *onto:uses* a service.

- *CQ3: Which interactions execute service X?*

  Figure 4.7 presents the SPARQL query which shows the interactions that execute a service. This is achieved by linking interactions to the service executed in those interactions (:executes). It is also possible to observe the individual interactions related to a sequence.

Overall, the ontology has no design anomalies as presented by the OOPS! tool and it addresses the requirements related to the competency questions defined.

[2]https://graphdb.ontotext.com/

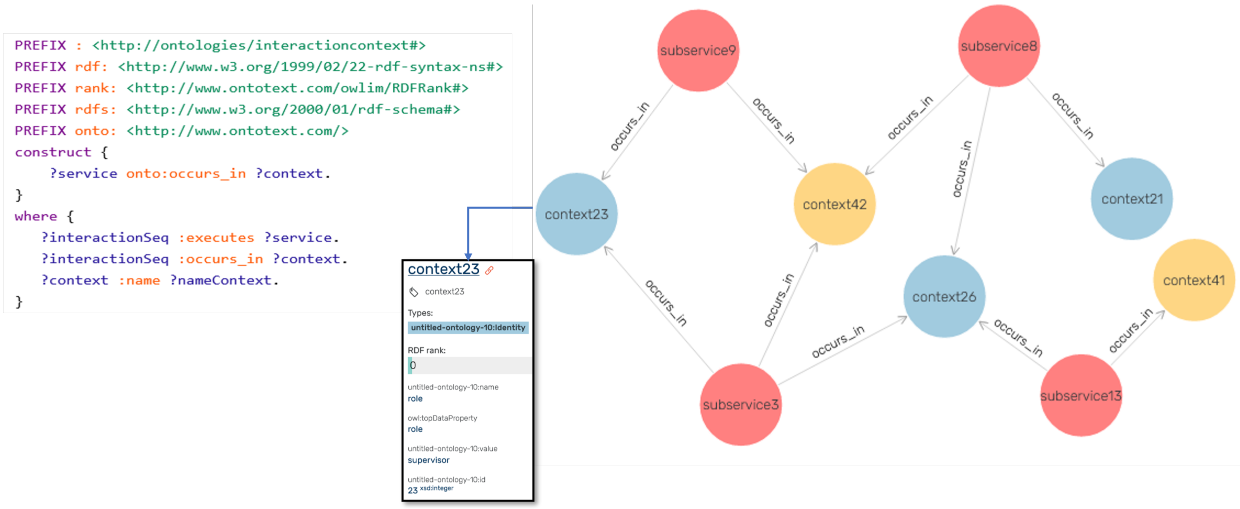


Figure 4.5: CQ1: In which contexts is a service executed?

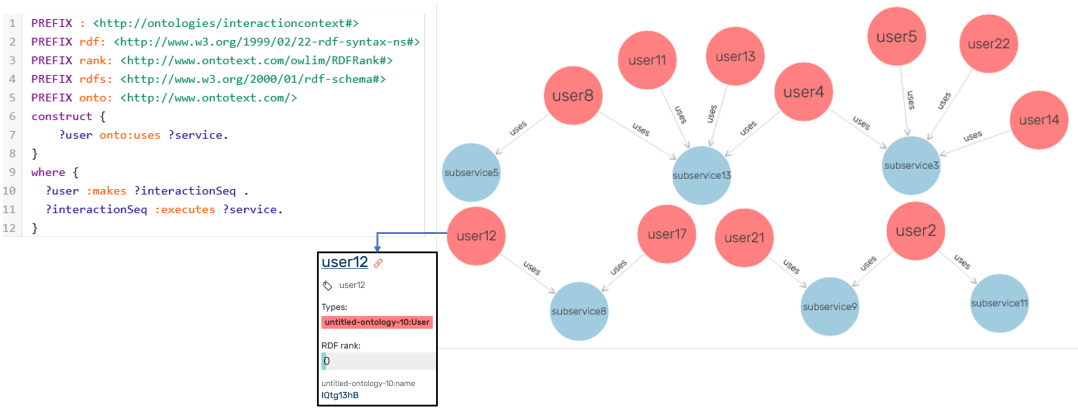


Figure 4.6: CQ2: Who are the users of a service or its sub-services?

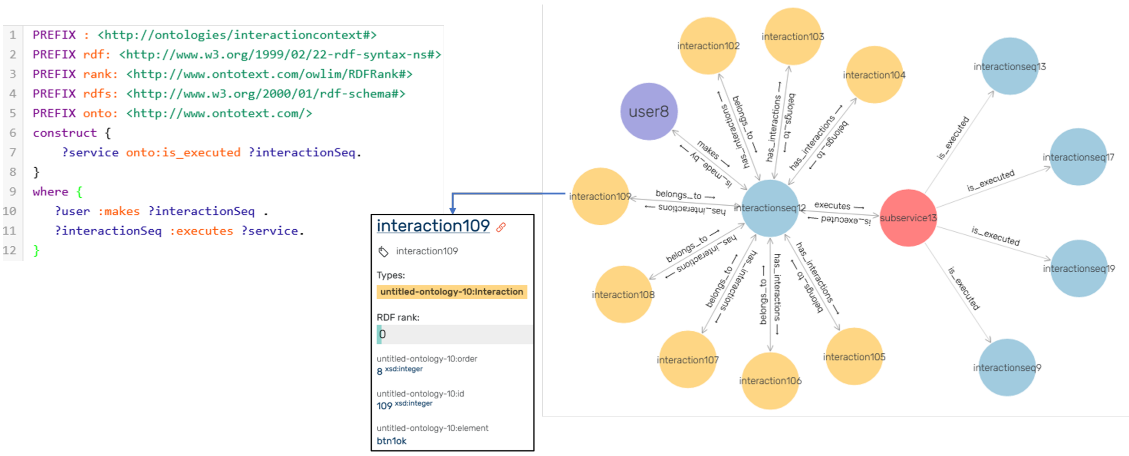


Figure 4.7: CQ3: Which interactions execute service X ?

4.1.3 Reasoning

The reasoning stage involves processing user and contextual data obtained from the use of smart products. This data is then used to create personalized user interfaces that adapt to each individual. In this context, recommendation systems is an approach that can leverage data for this purpose.

As presented in Figure 4.1 (Stage 3), the reasoning stage comprises two main steps: context pre-filtering and rating calculation, followed by collaborative filtering that utilizes user ratings. While user ratings are commonly obtained in traditional CF scenarios (e.g., star ratings in e-commerce), they could prove more challenging to achieve in S-PSS. This is because S-PSS ratings are primarily derived from user interactions (e.g., clicks, page views, etc.), reflecting the level of engagement with the digital services provided by the product. This approach allows for the generation of implicit ratings, used to uncover similarities between users and/or services.

In this work, the implicit ratings generated through these interactions are employed for the recommendation system algorithm. The process of these steps is detailed in the following subsections.

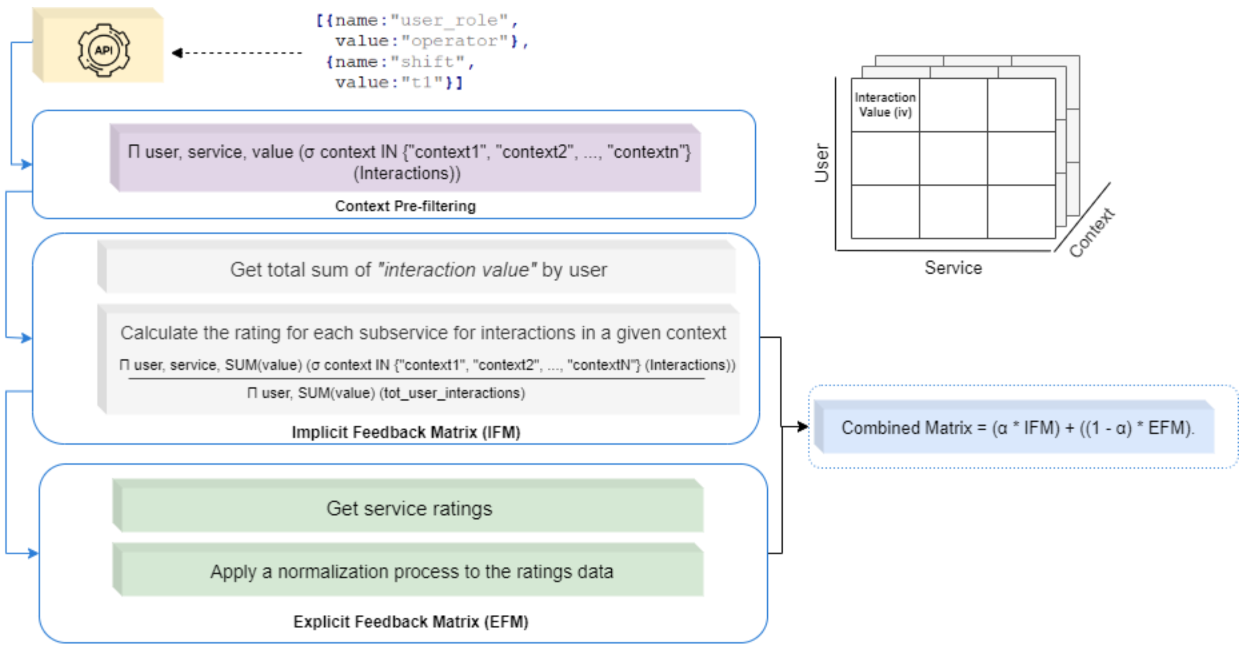


Figure 4.8: Rating calculation

Context Pre-filtering and Rating calculation

To achieve context-awareness in the framework, the first step involves assigning contextual attributes within the S-PSS to broader categories, a process also referred to as 'context generalization' (Smirnov et al., 2017). As depicted in the ontology (Figure 4.3), four contextual sources have been defined: *identity*, *device*, *environment*, and *time*. From these sources, contextual values can be grouped. For instance, specific user demographics (such as age-group or job type) can then form a broader category known as 'user profile.' Similarly, specific time periods can be combined to represent broader categories like 'season' or 'shift' (see Figure 4.9). Generalizing contextual attributes in this way reduces the dimensionality of the contextual space and mitigates data sparsity. However, this process is not automatic for the framework and requires an analysis for each individual S-PSS.

Once the contextual sources are selected and generalized, this work adopts contextual pre-filtering, known for its efficiency in narrowing down the data pool and providing accurate recommendations based on contextual data (Smirnov et al., 2017). In this approach, context awareness is achieved by filtering the input of the classical recommendation algorithm, followed by collaborative filtering technique. Context pre-filtering uses contextual attribute values as constraints for selecting ratings in the user-item space of a recommendation system (Kulkarni, Rodd, 2020).

As presented in Figure 4.8, the designed RS API receives a set of generalized contextual data in a JSON format and a SPARQL query is then 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 (Figure 4.9). Therefore, the filter parameters used in the query were non-exclusive to prevent significant reduction of 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 through direct input from users and directed to a specific service or item. The framework considers each type of interaction as a form of implicit feedback, and it is 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 of a user across a given set of contexts to the number of interactions with an individual service. In relational algebra this can be expressed as:

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

This provides a normalized measure of user activity that takes into account a broader range of user interactions, which allows for more accurate comparisons between users and services. As previously stated (see Section 4.1.2), interactions can occur in multiple contexts. Therefore, if the same interaction occurs in more than one context within the set 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, 2016). This approach can help with ‘cold-start" when there is no relevant information for a specific set of contexts and reduce the data sparsity problem associated with context pre-filtering. The data becomes less sparse as each interaction is associated with multiple ratings, which increases the probability that services will be accurately represented in the data. Moreover, more data points can be generated with this approach, which can help to mitigate sparsity by providing

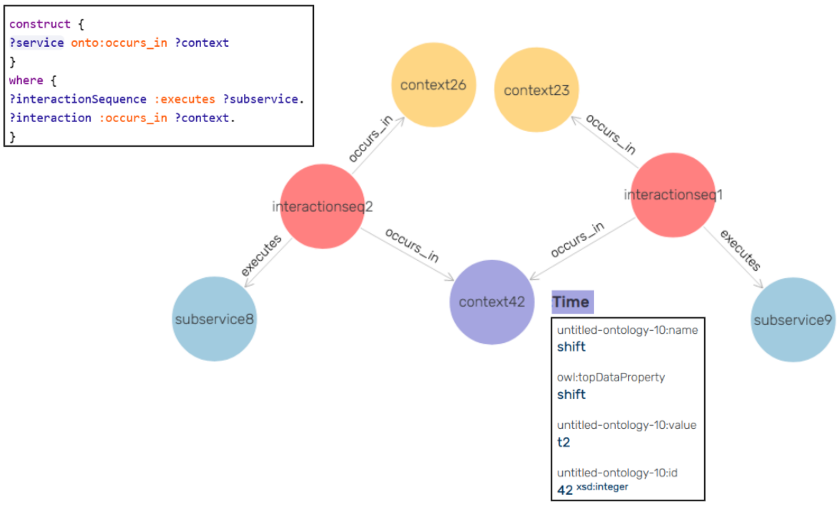


Figure 4.9: Querying interaction sequences occurring in several contexts

additional information that can be used to better estimate the true value of the interactions. Algorithm 1, sets out a high level overview of the process of calculating implicit ratings.

On the other hand, when users provide explicit feedback, a score is assigned 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 are averaged and 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 Matrix (EFM) to generate a combined matrix that incorporates weighted contributions from each type, as shown in Figure 4.8.

Collaborative Filtering

Collaborative Filtering (CF) is a widely used recommendation system technique that exploits available data containing user preferences and ratings to identify similarities between users or 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).

The K-Nearest Neighbors (KNN) algorithm is employed in this step. KNN is widely used 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 the user preference for a specific service (Kulkarni, Rodd, 2020). To implement the KNN algorithm, the matrix containing the computed ratings resulting from both user interactions and explicit feedback serves as the input.

To compute the similarities between users for predicting a service recommendation, three

Algorithm 1 Calculate ratings implicit

```
function CALCULATERATINGSIMPLICIT(users, context_list)
    ratingsImplicit ← []
    for user_id in users do
        total_interactions ← EXECUTE_SPARQL_QUERY(SELECT SUM(?valueInteraction)
        WHERE ?user a :User.
        ?user :makes ?interaction.
        ?interaction :type "INTERACTION"
        ?interaction :occurs_in ?context.
        ?context :value ?contextvalue.
        FILTER (?contextvalue exists context_list && ?user = user_id) )

        binding_set ← EXECUTE_SPARQL_QUERY (
        SELECT ?user ?subservice
        (SUM(?valueInteraction)/total_interactions AS rating) WHERE
        ?user :makes ?interaction.
        ?interaction :executes ?subservice.
        ?interaction :type "INTERACTION"
        ?interaction :valueInteraction ?valueInteraction.
        ?interaction :occurs_in ?context.
        ?context :value ?contextvalue.
        FILTER (?contextvalue exists context_list && ?user = user_id)
        GROUP BY ?user ?subservice)

        if binding_set is empty then
            return null
        end if
        for b in binding_set do
            ratingsImplicit.add({user: b.user, subservice: b.subservice, rating: b.rating})
        end for

    end for

    return ratingsImplicit
end function
```

similarity metrics found in the literature were evaluated: cosine, Euclidean, and Pearson correlation factor. These metrics are commonly used 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, Campos et al. (2010) employed the Pearson coefficient and variations of the KNN algorithm, together with an ad-hoc strategy, to take into account the temporal context for movie recommendations. 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 recommendation system to recommend e-learning resources to learners.

Distance Metric	Equation
Cosine Similarity	$c_{u_1,u_2} \frac{\sum_{i \in S_{u_1} \cap I_{u_2}} r_{u_1,i} \cdot r_{u_2,i}}{\sqrt{\sum_{s \in S_{u_1}} r_{u_1,i}^2} \cdot \sqrt{\sum_{s \in S_{u_2}} r_{u_2,i}^2}}$
Euclidean Distance	$d_{u_1,u_2} = \sqrt{\sum_{s \in S_{u_1} \cap S_{u_2}} (r_{u_1,s} - r_{u_2,s})^2}$
Pearson Correlation Coefficient	$r_{u_1,u_2} = \frac{\sum_{s \in S_{u_1} \cap I_{u_2}} (r_{u_1,s} - \overline{r_{u_1}})(r_{u_2,s} - \overline{r_{u_2}})}{\sqrt{\sum_{s \in S_{u_1} \cap S_{u_2}} (r_{u_1,i} - \overline{r_{u_1}})^2} \sqrt{\sum_{s \in S_{u_1} \cap S_{u_2}} (r_{u_2,i} - \overline{r_{u_2}})^2}}$

Table 4.1: Equations for Distance Metrics in Collaborative Filtering

Table 4.1 sets out the equations to calculate the three similarity metrics. $r_{u1,s}$ and $r_{u2,s}$ represent the ratings of users $u1$ and $u2$ for service s. $\overline{r_{u1}}$ and $\overline{r_{u2}}$ are their respective average ratings, and $S_{u1,u2}$ refers to the set of items that have been rated by both users. Using these similarity metric coefficients, we incorporated the kNN algorithm. The 2D combined matrix that represents a weighted contribution of both implicit and explicit ratings for each service was employed as input. Then, the neighborhood was limited to predict the weighted average rating of service s using the rating given to s by users most similar to u and returning the most relevant options k, ranked by the calculated rating:

$$predictedRating = \frac{\sum(sim(u,s) * r_{u,s})}{\sum sim(u,s)} \quad (1)$$

4.1.4 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.

Dimensions of the Design of Adaptive User Interfaces (AUIs)

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 4.2 and elaborated upon below.

I. **Adaptation Target**: Establishing adaptation targets is a fundamental step in digital platforms within S-PSS . The primary objective of customizing the UI based on outputs from the recommendation engine involves modifying layout elements (e.g., messages, cards, and buttons) which represent recommended services or smart device features, particularly 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

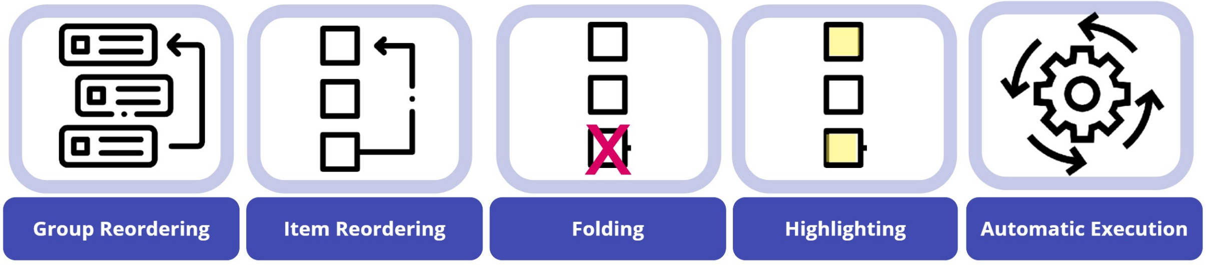


Figure 4.10: Adaptation Styles

styles, play a role in the visual presentation of the selected UI targets (Figure 4.10). 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, Schraefel, 2005).
- **Item/Service Reordering**: 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 AdaptUI is a recommendation-based and context-aware framework. AdaptUI 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 is retrieved, and (ii) when contextual data changes.

IV. Control over the adaptation To analyse 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 AdaptUI 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 4.1.3. 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 "reordering" 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.

Technical perspective and data schema for Adaptive User Interfaces

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. Unlike user interface description languages that are platform-dependent (i.e. Java FXML, Swift) or have steep learning curves (Nalpon et al., 2022), this work adopts a flexible UID data schema offering a good level of abstraction.

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 will be represented as a UID, guiding the translation of recommendation engine outputs into specific UI components.

Dimensions	Description
Adaptation Target	Identifies the specific interface elements or aspects for adaptation. For instance, 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. • Automation: The ability of the system to adapt without user input. • Participation: User engagement in adaptation, from minimal to extensive. • Visibility: Transparency in how and why adaptations occur.
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 in contextual information.
Evaluation of the adaptation	Involves assessing the effectiveness of adaptive changes through user feedback, performance metrics.

Table 4.2: Dimensions and Descriptions for Adaptive User Interfaces in Smart Product Service Systems. Adapted from (Oestreich et al., 2022)

The proposed framework employs a Server-driven UI design pattern to build the AUI, where UIDs are sent from the server to the client for rendering. This approach offers flexibility by allowing the server to generate the descriptors and dynamically adapting them based on user context and preferences (ThoughtWorks, 2022). The frontend API, depicted in Figure 4.1, receives recommended services (see Section 3.3.1) and generates the descriptors for the S-PSS UI based on the schema. This enables real-time adaptability of user interfaces in S-PSS applications without needing to redeploy. 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. It also manages user inputs (e.g., clicks or swipes), transmitting them for server processing. The UID generation schema is illustrated in Figure 4.11.

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 (Mabuyo, Sullivan, 2022).

The IUIElement interface captures common properties of UI elements such as idElement, label, type, but can also receive serviceInformation. The IUIContainer interface represents

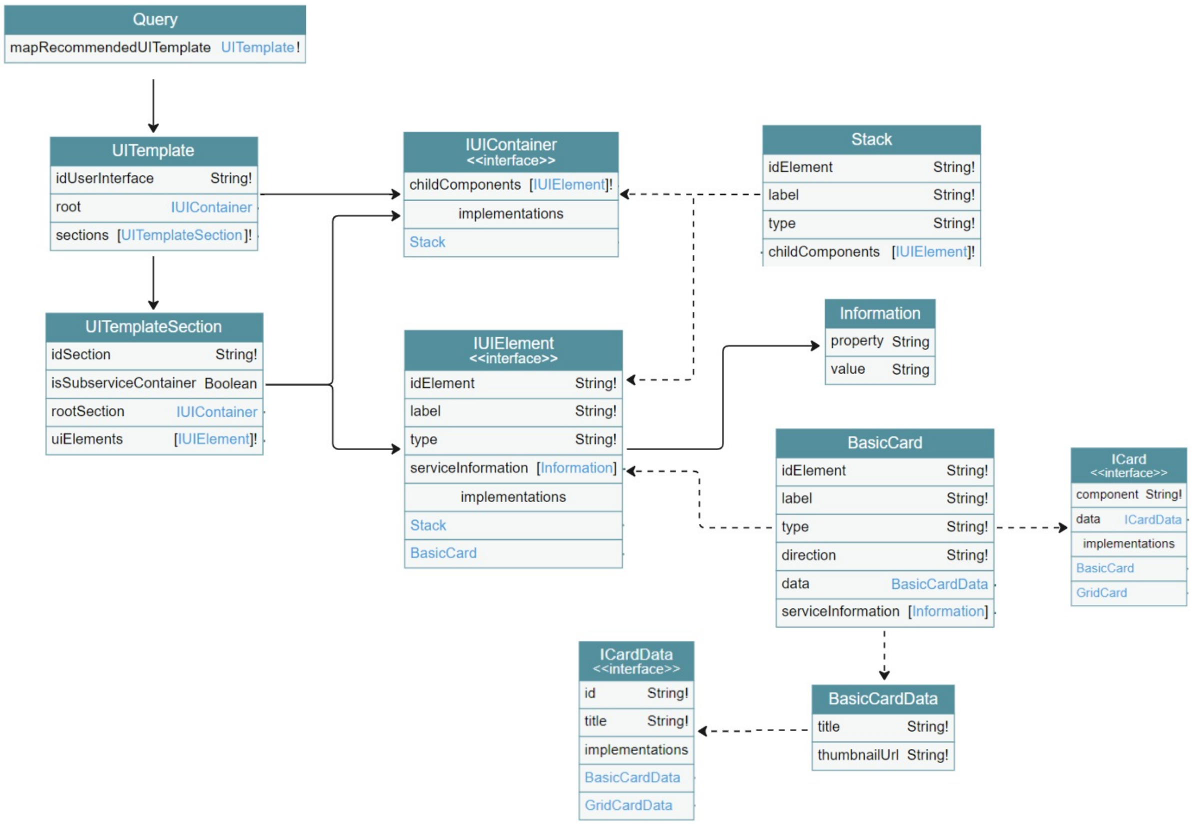


Figure 4.11: Schema Server Driven UI

containers or panels holding child UI components. The UITemplate type defines a template structure with an idUserInterface, a root container (IUIContainer), which represents a panel that can have multiple elements; and an array of template sections (UITemplateSection). The UITemplateSection represents a section within a UI template, which can hold multiple UIElements. Specific implementations can be created from these abstract types. For instance, Stack is a concrete type of container or panel that is both IUIElement and IUIContainer. BasicCard is a concrete implementation of an IUIElement. 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), subservices (a list indicating the 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 4.12 provides an illustrative example of the schema. Consider an application dedicated to smart wristbands, in which one of the key e-services is "User Monitorization." 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

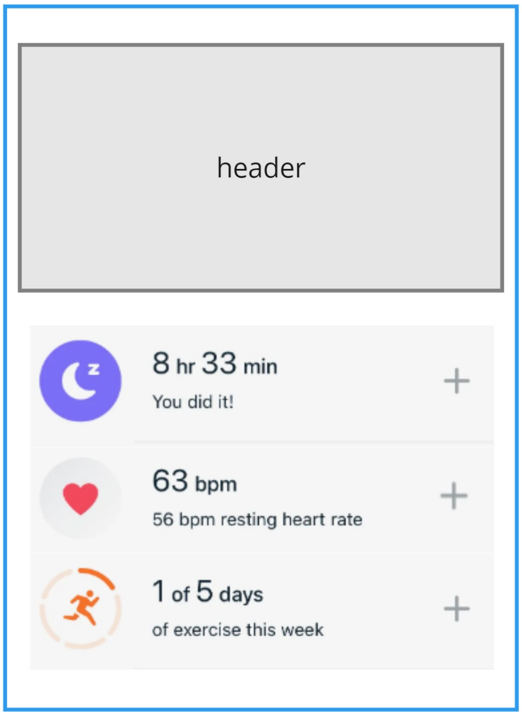


Figure 4.12: UI User Monitorization for smart wristband

morning for a user: "sleep monitor", "heart rate monitor", and "exercise monitor".

The query method "mapRecommendedUITemplate" takes recommended service and sub-service as parameters in JSON format, and navigates the hierarchy of e-services and e-subservices to identify the associated template and map the UI elements. The algorithm for this process is presented in Algorithm 2, 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 result from this method, is a template in a JSON format which is presented in Figure 4.13

4.1.5 **Monitoring & Evaluation**

The monitoring process of the framework highlights the evolution of the S-PSS. Feedback about decisions made by the application from surveys and user behaviour is monitored and evaluated. Three types of evaluation are considered: Offline Evaluation, Online Evaluation, and UX Quality Evaluation. This process also contributes to testing the hypotheses outlined in Chapter 1.

Algorithm 2 Map Recommended Services to UI Template

```
function mapRecommendedUITemplate(recommendedService)
    // Get the service template based on the recommended service
    serviceTemplate ← getServiceTemplate(recommendedService.service)

    // Get the parent section (subservice container) from the service template
    parentSection ← getSubserviceContainer(serviceTemplate.sections)

    for each subservice in recommendedService.subservices do
        subserviceTemplate ← getSubserviceTemplate(subservice)
        // Iterate over sections in the subservice template
        for each childSection in subserviceTemplate.sections do
            for each element in childSection.uiElements do
                targetContainer ← childSection.rootSection ? childSection.rootSection :
                    parentSection.rootSection
                // Add the element to the target container
                targetContainer.childComponents.push(element)
            end for
            // If the child section has a root, add it to the root parent
            if childSection.rootSection then
                parentSection.rootSection.childComponents.push(childSection.rootSection)
            end if
        end for
    end for
    // Return the modified service template
    return serviceTemplate
end function
```

Offline evaluation

Offline evaluation involves assessing the performance of the recommendation component based on historical data without directly involving users in real-time (Herlocker et al., 2004). Metrics are often used to measure the accuracy and effectiveness of recommendations. Table 4.3 describes three metrics, the first of which is Mean Average Error (MAE), a popular metric in recommendation systems which assesses rating prediction performance by calculating the average absolute difference between predicted ($\hat{y}_i$) and actual (y_i) user ratings in the test set . A lower MAE indicates better prediction performance, as it suggests smaller average errors in rating predictions.

Precision@k is a widely-used recommendation system metric that quantifies the accuracy of the top k recommended items for a user by determining the ratio of relevant items—those the user interacted with or liked in the test set in the top k recommendations.

In cases in which the order of recommendations needs to be evaluated, Mean Reciprocal Rank (MRR) measures the effectiveness of the recommendation by calculating the average reciprocal rank of the first relevant item. The rank (Rank_i) is the position of the first relevant

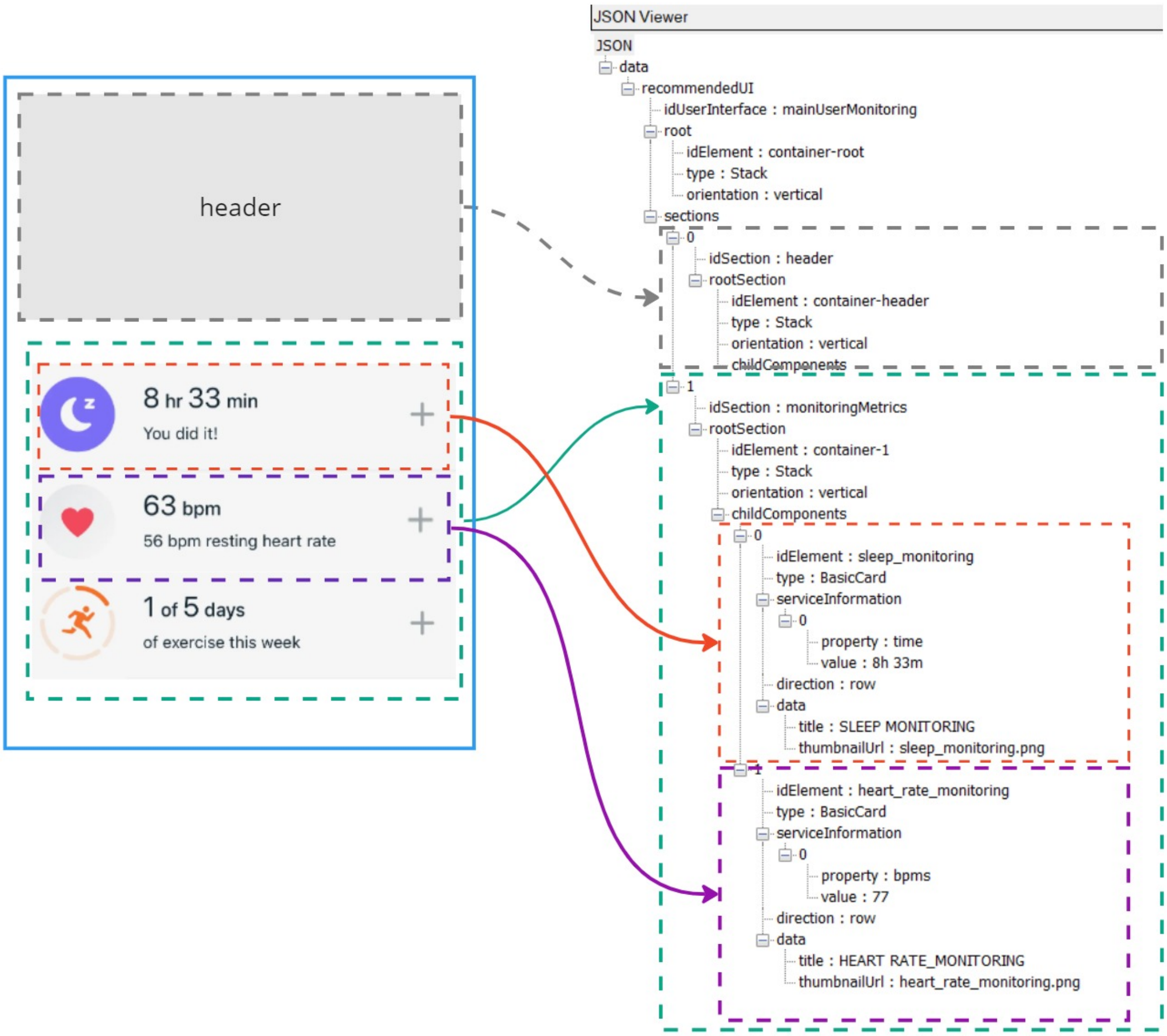


Figure 4.13: Response from method mapRecommendedUITemplate in JSON format

item in the list of recommendations.

Metric	Formula
MAE	$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert\hat{y}_i - y_i\rvert$
Precision@k	$Precision@k = \frac{\lvert\text{Relevant items in top } k\rvert}{k}$
MRR	$MRR = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\text{Rank}_i}$

Table 4.3: Offline evaluation metric Formulas

Online evaluation

Online evaluation involves the real-time assessment of system performance as users interact with the platform. In the present framework, two important metrics are considered: the Click Through Rate (CTR) and the Time-on-task. CTR is a metric used to measure the effectiveness of recommendations presented to users. It represents the ratio of users who engage with a specific recommendation to the total number of users who were exposed to that recommendation. A higher CTR indicates a higher level of user engagement with the presented recommendations, suggesting that these are relevant and useful. The framework measures the frequency with which users successfully accept the recommended services.

The average time-on-task is a metric that calculates the mean duration of time users spend completing a specific task or set of tasks. It provides an overall measure of the efficiency of task completion across all users. The formulas of each of these metrics are set out in Table 4.4.

Metric	Formula
Click Through Rate (CTR)	$\text{CTR} = \left(\frac{\text{Number of Clicks}}{\text{Number of Impressions}}\right) \times 100$
Time-on-task	$\text{Time-on-task} = \text{End Time} - \text{Start Time}$
Average Time-on-task	$\text{Average Time-on-task} = \frac{\text{Total Time-on-task for all Users}}{\text{Number of Users}}$

Table 4.4: Online evaluation metric Formulas

UX Quality Evaluation

The UX Quality evaluation of the framework focuses on the more pragmatic aspects of the experience. An evaluation questionnaire is used for this purpose, which consists of a set of constructs and the participating questions for each construct. 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 to evaluate the quality of the UX. Table 4.5 presents the constructs alongside a brief description. Each construct is divided into different subclasses that break down the evaluation process into more specific categories.

Construct
Quality: Refers to the perceived performance and accuracy of adaptations suggested by the recommendation component and considers the following sub classes: - **Accuracy**: The degree to which users feel that the recommendations align with their interests and preferences - **Novelty**: The extent to which users receive new and interesting recommendations. - **Diversity**: Measures the variety of items in the recommendation list. - **Context Compatibility**: Evaluates whether recommendations consider general or personal context requirements.
Ease of use: Efficiency, or perceived cognitive effort, measures user capacity to swiftly and accurately complete tasks with ease and minimal frustration. - **Ease of initial learning**: Also known as “learnability”, assesses user ability to understand adaptations. - **Ease of decision making**: Extent to which decision-making processes are streamlined for user convenience.
Perceived Usefulness : Extent to which a user perceives that using the AUI improves performance, as compared to the experience with no help.
Interaction Adequacy : Ability of the system to elicit user preferences, enable user feedback, and explain the reasoning behind adaptations.
Transparency and Control - **Transparency**: Degree to which a system allows users to comprehend the logic behind its recommendations. - **Control**: Capacity of the system to enable preference adjustments and the extent to which users feel in control during interactions.
Behavioral Intentions : Extent to which users agree to use the system, their acceptance of the system and intention to return.

Table 4.5: Evaluation questionnaire constructs for recommendation-based adaptive user interface. Adapted from Pu et al. (2011) .

4.2 AdaptUI Implementation

Prior to exploring the case studies presented in the following chapters, it was necessary to develop the components that would allow testing the implementation. First, the employed software architecture was established, providing a high-level structure of the components and detailing how they would be interconnected. Furthermore, a roadmap outlining the planned development was also elaborated. Both of these aspects will be explained in the following subsections.

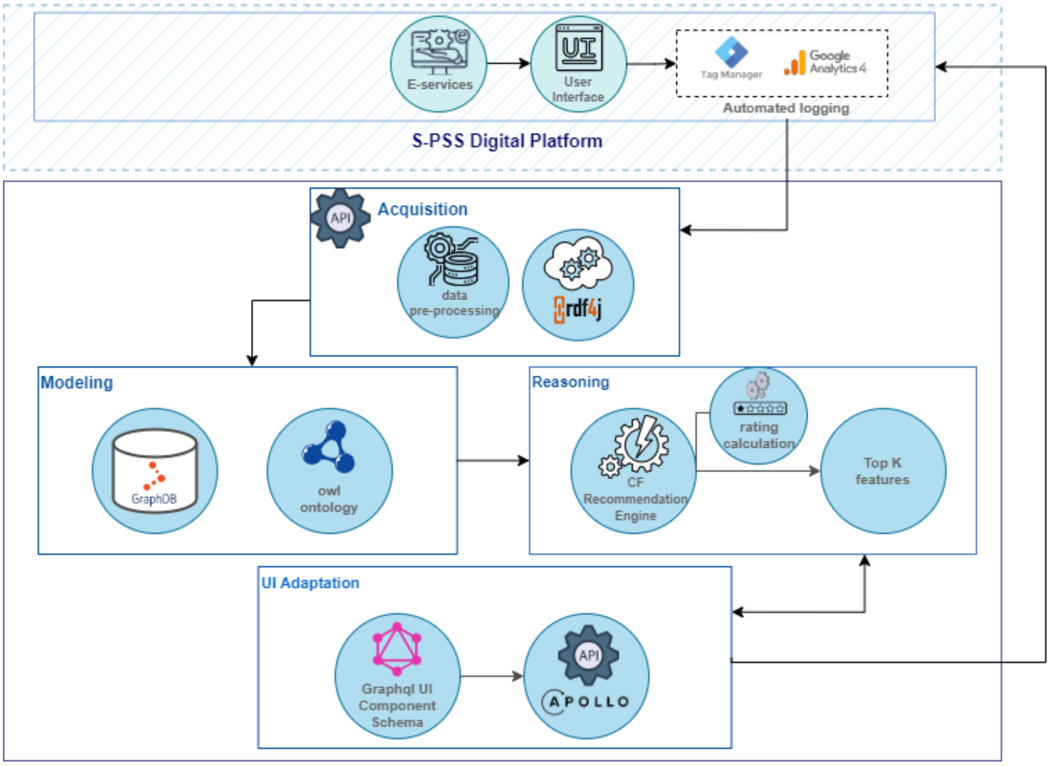


Figure 4.14: General Architecture for AdaptUI implementation

4.2.1 Software Architecture Implementation

The software architecture developed for implementing the framework is depicted in Figure 4.14. The digital platform provided by the S-PSS offers electronic services, leveraging connectivity with the smart products. Users access these services through a UI, available on web-apps, mobile apps, or directly on the product. This is an external entity to the framework, where the framework is deployed. As explained in Section 4.1.1, interactions are logged using an automated method, employing Google Analytics 4 (GA4) and Google Tag Manager, with data stored in the Big Query service. GA4 is a widely used web analytics service that effectively tracks and reports website traffic and user interactions.

The implementation follows a Service-Oriented Architecture (SOA) with loosely coupled, independent services communicating through well-defined APIs. The "Data Importer" component is a Java-based API using the RDF4J library[3], a robust framework for working with RDF data. RDF4J supports W3C standards related to RDF, SPARQL, and other technologies, facilitating the creation, parsing, serialization, and complex queries of RDF data. It also provides interfaces for working with RDF repositories, allowing efficient storage and retrieval of RDF data.

The "Data Importer" serves a dual purpose in the framework's data acquisition process. Captured interactions undergo pre-processing before being uploaded in batches using this

[3] https://rdf4j.org/

component. Additionally, the "Data Importer" receives contextual information. Figure 4.15 depicts the methods employed for storing information, showcasing how the API handles and stores the data in GraphDB. GraphDB is a graph database that strictly adheres to RDF and SPARQL standards, serving as the repository for the ontology used as a model within the framework.

Method	Description
`insertBatchContext(ArrayList<ContextInteraction> is, String GRAPHDB_SERVER)`	Inserts a batch of Context objects.
`insertBatchService(ArrayList<Service> is, String GRAPHDB_SERVER)`	Inserts a batch of Service objects.
`insertBatchSubservice(ArrayList<Subservice> is, String GRAPHDB_SERVER)`	Inserts a batch of Subservice objects.
`insertBatchUser(ArrayList<User> is, String GRAPHDB_SERVER)`	Inserts a batch of User objects.
`insertContext(ContextInteraction is, String GRAPHDB_SERVER)`	Inserts an individual Context object.
`insertInteraction(ArrayList<Interaction> is, String GRAPHDB_SERVER)`	Inserts a batch of Interaction objects.
`insertInteractionSequence(ArrayList<InteractionSequence> is, String GRAPHDB_SERVER)`	Inserts a batch of InteractionSequence objects.

Figure 4.15: Methods Data Importer

Similarly, the same RDF4J API is also used to connect the recommendation system, through the data stored in the ontology and query the recommendations, the methods employed to this end are presented on Figure 4.16.

Method	Description
`calculateRatings(ArrayList<ContextInteraction> contextList, String GRAPHDB_SERVER, boolean isExclusive)`	Calculates ratings based on a list of context interactions.
`getRecommendationService(String userid, boolean on, int topk, String GRAPHDB_SERVER)`	Retrieves service recommendations based on specified parameters for a given user.
`getRecommendationSubService(String userid, boolean on, int topk, String service, String parameter, String GRAPHDB_SERVER)`	Retrieves sub-service recommendations based on specified parameters for a given user.
`GetInteractionByContextValue(String GRAPHDB_SERVER, ArrayList<ContextInteraction> contextList)`	Retrieves sequence interactions based on a list of context values from a GraphDB server.

Figure 4.16: Methods Recommendation System

For UI adaptation, the frontend API, developed using Apollo GraphQL [4], employs a robust GraphQL schema. Apollo GraphQL was chosen for its ability to concisely articulate the application data model and supported operations. Serving as a contract between the client and server, it defines how data can be requested and the expected response in JSON format. The schema detailed in Section 4.1.4, generates a UID based on recommendations, dictating the UI

[4] https://graphql.org/

elements to be rendered on the user application interface. The schema is available to download for further exploration [5].

4.2.2 A roadmap for AdaptUI Implementation

A roadmap for AdaptUI implementation is presented in the form of a matrix as depicted in Figure 4.17. The matrix establishes a connection between the dimensions essential for designing an AUI for S-PSS outlined in section 4.1.4, which are represented as rows. The columns correspond to the various phases or stages of the framework, as detailed in Section 4.1. The roadmap is a visual representation of the framework progression and aligns with the principles of 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 Figure 4.17, this pre-implementation stage is included, referred to 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.

This roadmap has been developed and refined based on the case studies studies presented in Chapter 5 and Chapter 6. The practical experiences detailed in these case studies have played a role in shaping and improving the AdaptUI implementation.

[5]https://github.com/aicarrera/AppMachine/blob/main/server-backenddrivenUI/src/schema/types.ts

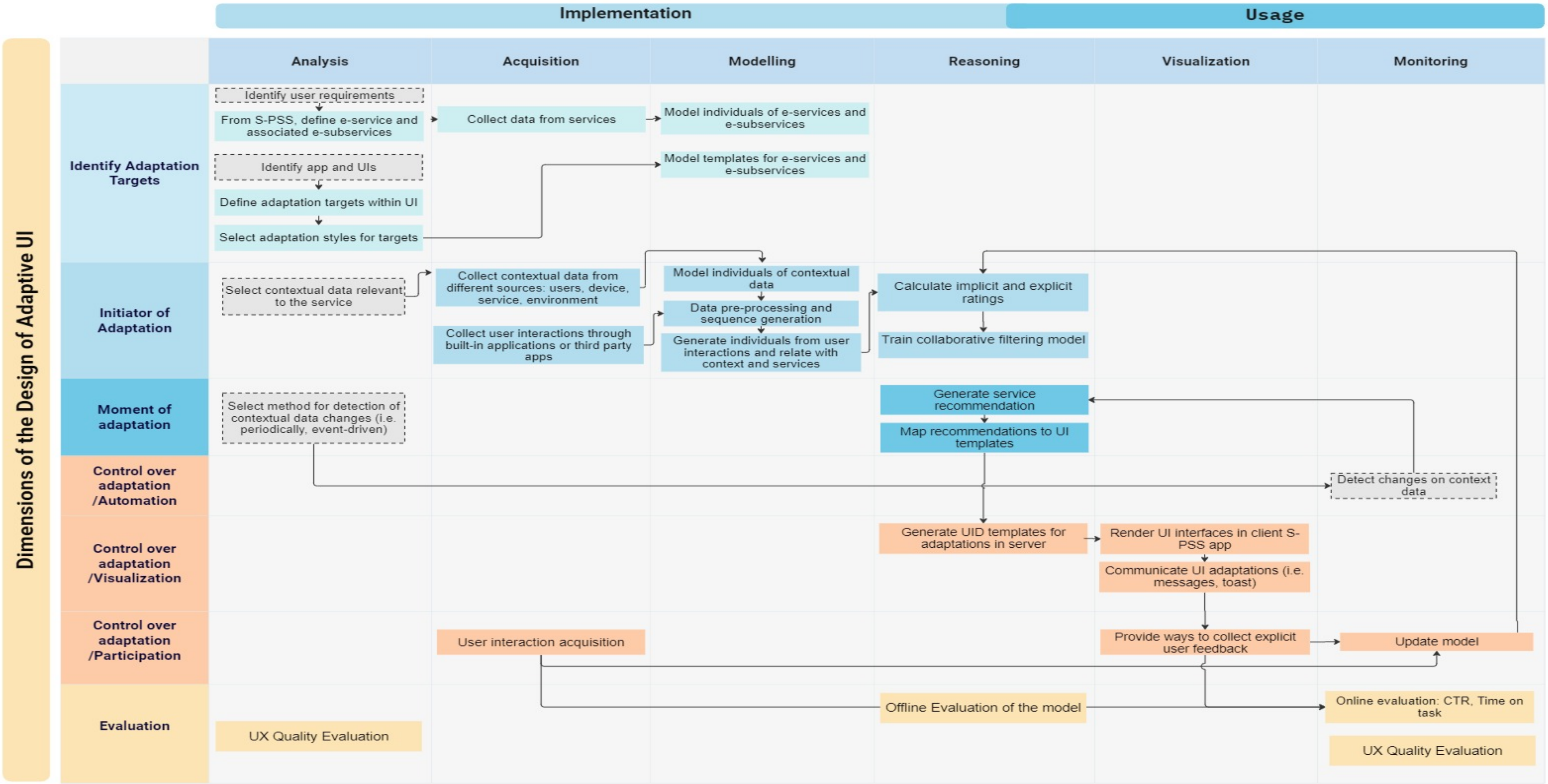


Figure 4.17: Roadmap for AdaptUI Implementation incorporating design and implementation dimensions

4.3 Framework Scope & Limitations

This section outlines the scope and limitations of the framework.

- In terms of modeling, the incorporation of entities relevant to the product accurately represents the “smart" capabilities of the product. However, it is important to note that the proposed framework focuses primarily on user interaction with a smart device. Therefore, any discussion of physical components and connectivity aspects can be considered limited, as it falls outside the scope of this thesis.
- The recommendation-based approach employing collaborative filtering clearly defines the scope of the types of adaptations that can be produced by the framework. These are limited to the presentation and content of the interfaces, and may not necessarily provide guidance to users through specific processes or actions. In this latter case, other approaches might be more suitable for step-by-step tasks. However, the recommendation block can be changed for another type of algorithm (e.g., a sequence-aware recommendation system) and the other blocks will still be able to produce an AUI.
- The framework primarily focuses on the structural and interaction aspects of AUI. Contextual data selection, which is highly dependent on the nature of individual S-PSS applications and user scenarios, is beyond the scope of this work. From the state of the art (see Section 3.2.4), context-based activity modeling is defined as a methodology that allows designers to represent services based on activities and contextual-related data (Kim, 2023).
- It is important to highlight that the detection of contextual data changes can be managed in a variety of ways. This is discussed in the background (See sections 2.3.3; 2.3.6) in terms of considering the responsibility and frequency of the context dissemination and acquisition as presented in Table 2.6. For instance, changes in contextual data can be periodically monitored and retrieved at predefined intervals (e.g., every X minutes). Another option is to employ an event-driven mechanism (Michelson, 2006), to instantly detect and notify the system of any contextual changes. The choice of implementation strategy will affect the moment in which adaptations occur.

4.4 Conclusions

This chapter introduces the AdaptUI framework, designed to fill existing research gaps in the context of S-PSS. The framework acts as a guiding structure for system development, focusing on the "usage" stage of the S-PSS lifecycle, in which digital platform user interfaces facilitate communication between users and smart products. The real-time user interface adaptation of the framework utilizes context-aware recommendation systems and exploits internal data sources, such as user interactions and contextual information.

The framework components align with the life cycle of context-aware applications, and thus digital services, sub-services, context, and their association with user interactions can all be modeled. User interactions are organized into sequences, providing a deeper understanding of user behavior. Additionally, user interactions are linked to contextual features, ensuring that the reasoning engine can calculate implicit ratings for services within each contextual feature.

The rating calculation process of the framework is fundamental, as it classifies interactions between explicit and implicit forms of feedback. Implicit feedback captures user selections and interactions with services, while explicit feedback involves the user explicitly evaluating a service. These ratings are fed into the KNN algorithm to achieve collaborative filtering recommendation. In collaborative filtering, the system identifies users with similar behaviors and recommends services based on the preferences of users with similar profiles. The KNN algorithm, in this context, uses the ratings generated to find the nearest neighbors, allowing the system to recommend services that align with the preferences of users who have demonstrated similar interactions.

The framework addresses data sparsity and cold start problems by handling contextual features as "inclusive". As such, contextual features do not constraint the data in this approach, and the result is that ratings are higher when multiple contextual features coincide.

To implement an AUI, the framework not only relies on reasoning engine results but also needs a component for the generation of the user interface. To this end, the framework employs a server-driven UI approach, supported by a data schema that links e-services and UI elements through template creation. This method ensures uniformity in the UI across diverse platforms and provides flexibility in generating UI descriptors aligned with recommendations. For successful implementation, it is necessary to ensure the definition and implementation of the components specified in the interface descriptors in the application client for rendering. Beyond technical considerations, the framework emphasizes diverse dimensions which are essential for AUI design.

For continuous improvement, the framework incorporates monitoring and evaluation. Offline evaluation is used to test the model in terms of precision and MAE, employing similarity metrics like cosine and Euclidean distances and the Pearson correlation factor. Online evaluation considers real-time data obtained from user interactions, capturing metrics such as time-on-task and the click-through rate. The UX evaluation focuses on user perceptions, with specific constructs tailored for recommendation systems, and provides a detailed assessment of the recommendation-based framework.

In the following chapters, two case studies are presented to better understand the implementation and components of AdaptUI.

5 Case Study: Smart Vending Machine

5.1 Introduction

The widespread adoption of home IoT technology has extended the application of S-PSS to smart appliances (Zheng et al., 2019b). This expansion enhances personalization, energy efficiency, and predictive maintenance solutions. S-PSS, seen as an IT-driven value co-creation strategy, promotes positive interactions through SCP capabilities, creating new e-services and maximizing value in tangible and virtual goods. In this case study, the selected application was a smart coffee vending machine. Major international manufacturers are becoming increasingly interested in the market growth potential of smart appliances (Yu, Sung, 2023).

The case study focuses on testing the implementation of the framework and validating each of its components. Utilizing user interaction data from an SCP, the framework dynamically adapts the UI in a web app. This adaptation takes into account the context and previous interactions of the users, with the aim of enhancing their overall experience with the machine.

In the subsequent subsections, the case study elements are examined. First, the case study is presented, followed by an analysis of the technical implementation of the framework and the AUI. Then, the evaluation process for both the recommendation component and UX is described. Finally, the conclusions section presents significant findings and the limitations of the study.

5.2 Case Study Set-up

The protocol followed for the case study is presented in Figure 5.1. For this study, a responsive web app was developed as the UI for the machine using the Next.JS [1] framework and Chakra UI [2] for the frontend of the app. Interactions of 45 users aged between 19 and 40+ years were used in this experiment, collected over a period of 50 days (see Table 5.1). Prior to participating,

[1] `https://nextjs.org/`
[2] `https://chakra-ui.com/`

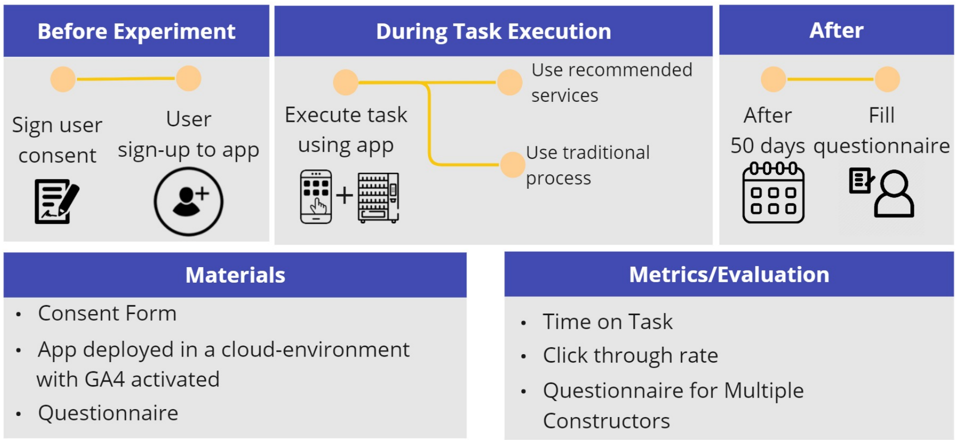


Figure 5.1: Protocol Case Study 1

each user formally consented to their interactions being recorded through the application. The users registered in the application, providing a username and selecting a role for themselves. During the experiment, the users executed the task of using the app to buy a beverage in the smart vending machine. A total of 10,608 interactions were recorded during the time period. All data was anonymized and does not contain sensitive user information.

The types of beverages obtained from the machine were considered as the services. These were registered in the ontology prior to the experiment, together with the type of information needed to deliver the service. In total 18 possible services were registered. Figure 5.2 presents part of the UI employed in the experiment. The initial screen, titled *Recommended for you*, is dynamically generated using user-specific contextual data and updated if these values change. The top 3 services recommended to the user are presented in the form of card components, which are constructed based on a user interface descriptor provided in JSON format (Figure 5.2a). This descriptor is generated using input from the recommendation engine, as well as information derived from the services specified in the ontology and the related information required to function effectively (See chapter 4.1.3).

However, users could also complete the process on the second tab (*Prepare your beverage*) (see Figure 5.2b), if they did not agree with the recommended services. The UI allowed the users to rate the recommended services, and thus gather explicit feedback.

Attribute	Item	Freq.	Attribute	Item	Freq.	Attribute	Item	Freq.
Age	19-25	2	Gender	Male	36	Occupation	Graduate student	11
	26-30	25		Female	9		Supervisor	12
	31-35	7		Non-Specified	0		Administration	5
	36-40	8					Researcher	9
	>40	3					Lecturer	8
Total Users:	45							

Table 5.1: Participant demographics

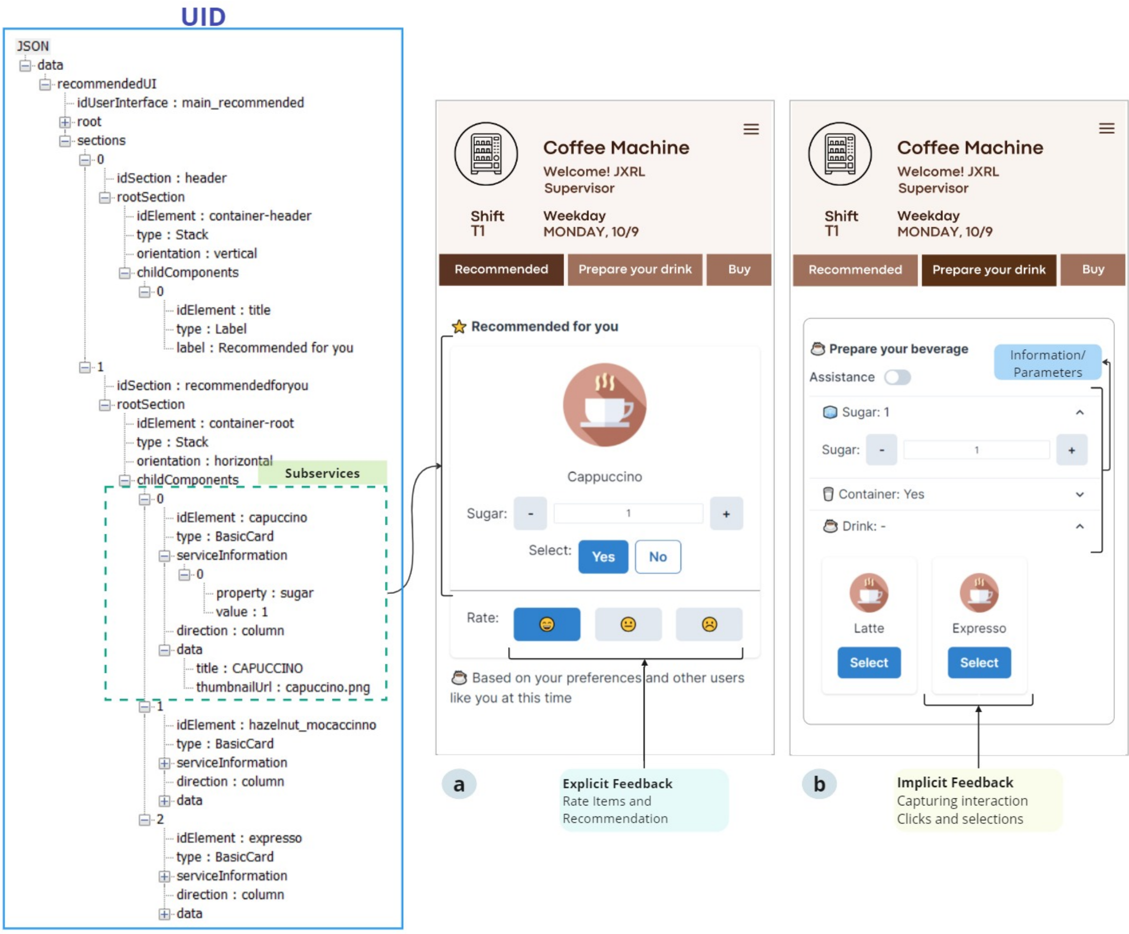


Figure 5.2: UI Smart Coffee Machine: (a)Options generated through the recommendation component and presented through an user interface descriptor. (b)Tab displaying options to prepare beverages

5.3 Technical implementation overview

For the data acquisition, user interactions on the app were captured using GA4. 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. Of the 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 an unique identifier and only interactions involving these elements were considered, which generated 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 as presented on Table 5.2. These actions included clicks on buttons and tabs, as well as session initiation. Table 5.3 presents an example of a valid sequence.

Figure 5.3 illustrates the flow of events involved in creating the AUI with service recommendations in the form of a sequence diagram. The user interface was developed using Next.js

Metric	Average Value
Sequence Length	7.82
Sequences per User	50.26
Most Used UI Elements	**Occurrences**
btn3sugardec_click	2176
btn1select_click	1357
btn1ok_click	1285

Table 5.2: Summary of User Interaction Data

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

Table 5.3: Example of an interaction sequence collected by the system

and React to create UI elements and the layout. In this particular case, the triggering events for generating recommendations are “user login" or changes in contextual data detected on the web app. Three types of contextual data are considered: *user role*, *shift*, *weekday,* which represent user and time.

To ensure a clear separation between the front-end and RS component, the framework employs separate connector APIs. This design choice ensures that the recommendation system is decoupled from specific UI implementations, promoting flexibility and enabling usage across a range of scenarios and with various UIs .

The frontend API developed in GraphQL connects to the RDF4J API which includes a collection of operations that use SPARQL statements to obtain and filter information from the ontology, as well as operations relevant to the CARS model. The functionality of the recommendation system component has already been explained on section 4.1.3.

When service recommendations are retrieved, they are returned to GraphQL. Based on the obtained data, these recommendations are transformed into UI elements using the descriptor. The back-end then constructs a response object and returns it to the front-end. The front-end, developed in Next.js, then renders the UI elements accordingly, presenting the recommended services to the user. In the frontend, these elements are implemented as self-contained snippets of code known as components.

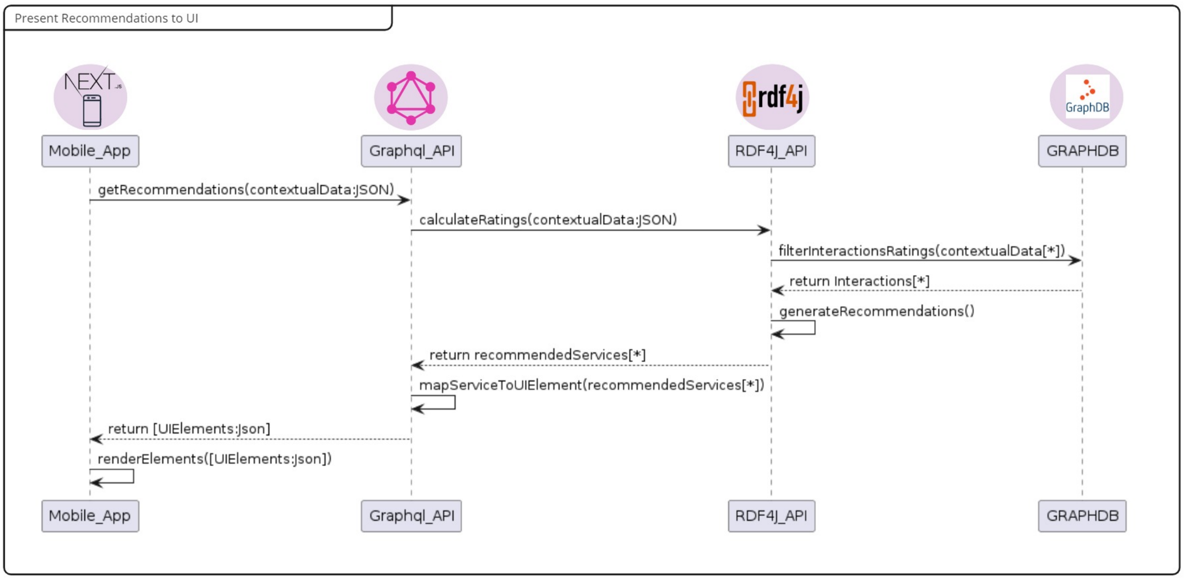


Figure 5.3: Sequence diagram illustrating the process of generating recommendations for a mobile app, including the Mobile App, APIs, RDF database, and UI rendering.

5.4 Analysis for the implementation of the Adaptive User Interface

In the process of generating the UI for the smart vending machine, the adaptation target was the *Recommended for you* e-service and the beverages presented to the user (see Figure 5.2). In this case, the recommended services are presented in a Card-based UI which are gaining popularity in many mobile and web apps (Rodrigues et al., 2017). Table 5.4 breaks down the analysis of the rest of design dimensions for UI adaptation in this 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	**Recommendation-based**		
	Automation System automatically identifies changes in context.	**Participation** User can accept or decline recommendation.	**Visibility** Cards with recommended services are shown to user.
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 5.4: Dimensions considered in the case study

From the technical perspective, Figure 5.4 depicts the GraphQL schema definition, used to create the AUI for recommendation services. The goal is to dynamically generate and adapt visual elements. The schema defines a set of interfaces and types that represent the components and data structures of the user interface. A *Card* is a self-contained component that can display information, but also contains other elements. The ‘*ICard*‘ interface serves as a base interface for all types of cards in the UI, such as ‘*BasicCard*‘ and ‘*GridCard*‘. Each card type implements the ‘*ICard*‘ interface and has specific fields and data structures associated with it. The ‘*ICardData*‘ interface defines common data fields for the card data, like ‘*id*‘ and ‘*title*‘, which are then implemented by types such as ‘*BasicCardData*‘ and ‘*GridCardData*‘, adding additional fields specific to each card type. The ‘*HStack*‘ type represents a horizontal stack of cards and contains a field ‘cards‘ that possesses an array of ‘*ICard*‘ components. In this way, multiple cards can 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 ‘*HStack*‘ with the recommended cards. The schema defines a 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 to be able to interpret them and generate the appropriate visual elements and layout. Additionally, the client side can manage user interactions and communicate them back to the server for processing.

The advantage of this approach is that the implementation of UI elements on the client side ensures that the system can consistently render and utilize the same schema across multiple platforms. This promote reusability, extensibility, and ease of development.

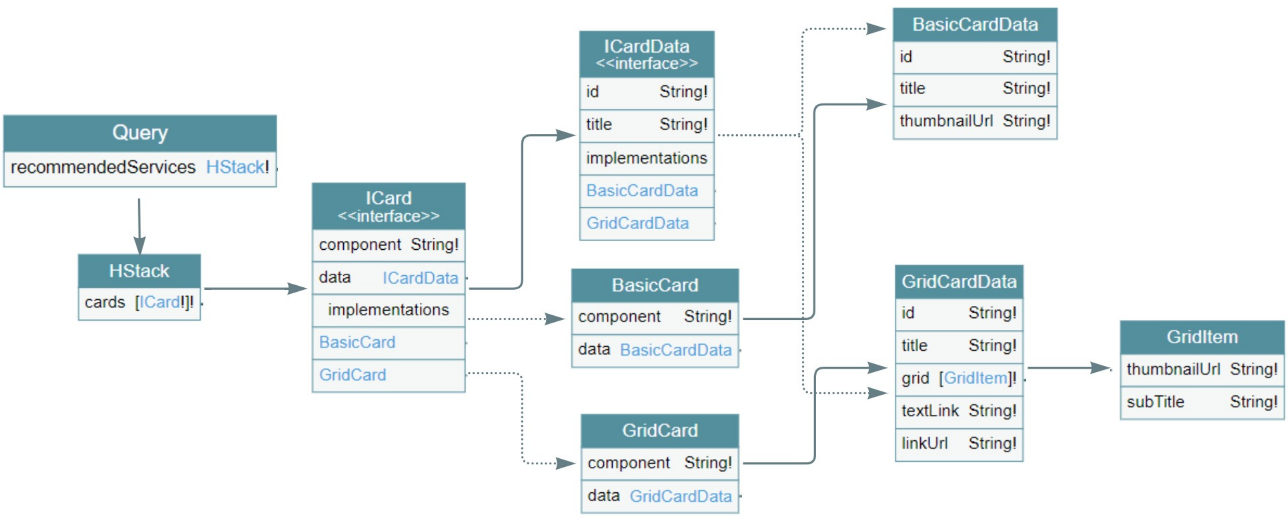


Figure 5.4: GraphQL Schema for Recommended Services

5.5 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.

5.5.1 Recommendation Component Evaluation

The effectiveness of the recommendation component of the framework was evaluated with an offline assessment using the collected dataset (Herlocker et al., 2004). An offline assessment involves the evaluation of the recommendation system performance based on historical data, in this case pre-existing user interactions.

The evaluation consisted of a series of experiments comparing the 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). MAE, on the other hand, is another popular metric, which 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 4.1.3, the interactions captured were treated as implicit feedback. For explicit feedback, users had the in-app option to express their opinion on the recommended items by rating them with emoticons. The available emoticons were 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 $\approx$7% of the total number of interactions.

To provide service recommendations to users in S-PSS, a straightforward assumption is that users will prefer services they have recently selected or rated, since familiarity reinforces trust in recommendation (Pu et al., 2012). This *naive* approach is easy to implement and leads to the presumption that user behavior could be relatively constant when using smart products. In the present study, it was used to obtain and present the information required for the service (e.g. sugar level).

However, this approach has some limitations as it does not take into account the individual context of the user, which can vary over time, leading to poor quality recommendations. 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 in our case study, indicating that this simple method may have some effectiveness in recommending services to users.

Each set of experiments was repeated 20 times using randomly selected training (70%) and test sets (30 %), 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 employing the Pearson correlation factor, and cosine and Euclidean distance metrics. The precision@K scores were evaluated at k=1, 2, and 3, as depicted in Figure 5.5.

On average, the kNN with cosine distance performed slightly better than the Pearson. There were some noticeable differences in the effectiveness with Euclidean distance at k=2: the median score was 0.57, and for 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 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 each method to the baseline precision score of the naive method, the kNN models clearly outperformed the baseline. Of these, the kNN with cosine distance performing most consistently overall.

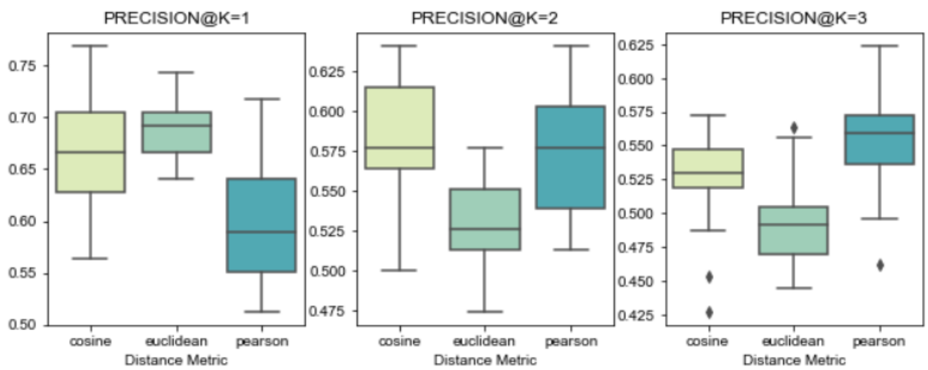


Figure 5.5: Precision@K results without using contextual data

The results of Precision@k using contextual data are presented in Figure 5.6, and in Table 5.5. These represent an improvement with the use of contextual data. At precision@1, on average Euclidean and cosine distances performed better. This could be due to effectively identifying the closest item to the user input, based on the shortest distance in the feature space. However, as k increases, Euclidean distance is outperformed by cosine and Pearson, possibly because it does not take into account the correlations between users beyond their pairwise distance (Ricci et al., 2011). On the other hand, the Pearson performance slightly outperformed the other distance metrics as *k* increased. This could be because of the sensitivity of the 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.

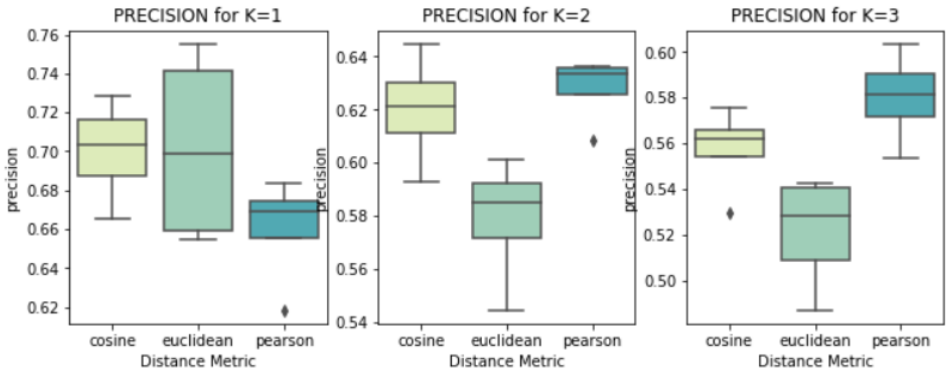


Figure 5.6: Precision@K results with Context

The average MAE results for the two experiments are set out in Table 5.5 . A lower MAE indicates better model performance in predicting rating values. The results reveal a slight improvement in the second test. This suggests that the model could better predict the rating values in the second test using contextual data.

After evaluation, we selected 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 between users beyond pairwise distance, cosine distance delivers a better understanding of user preferences and improves the accuracy of recommendations.

	No Context			With Contextual Data		
	Cosine	Euclidean	Pearson	Cosine	Euclidean	Pearson
Precision@1	0.665	0.689	0.600	0.700	0.702	0.660
Precision@2	0.583	0.529	0.573	0.622	0.575	0.628
Precision@3	0.527	0.492	0.556	0.556	0.517	0.587
MAE	0.114	0.133	0.102	0.110	0.123	0.094

Table 5.5: Case Study 1 – Results of Precision@K and MAE

5.5.2 Validation of the UX

The UX of the framework implementation was assessed following the protocol presented in Figure 5.1. Participants provided feedback through a questionnaire which was developed based on the user-centric evaluation framework proposed by Pu et al. (2011), previously described in Chapter 4.1.5. Feedback questions were developed based on the constructs predefined in the framework, and were adapted to ensure relevance to the S-PSS context.

The primary objective of the questionnaire was to gather valuable user feedback (see participants in Table 5.1) and evaluate several aspects of their experience, including behavioral

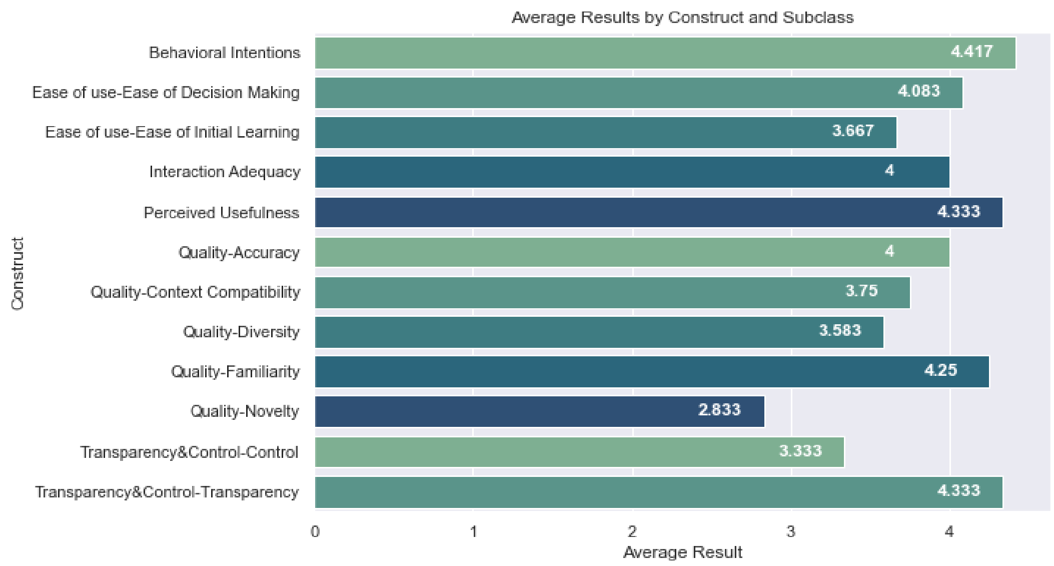


Figure 5.7: Case Study 1 - Results of UX constructs for evaluation of user perceptions

intentions, ease of use, perceived quality, beliefs, and attitudes to the S-PSS. To effectively capture user responses, a 5-point Likert scale ranging from "strongly disagree" (1) to "strongly agree" (5) was employed for each question. Figure 5.7 presents the results for each construct and sub-classification where necessary. The questionnaire and its results, presented in Table 5.6, provide a general view of user perceptions into various aspects of the UX with the AUI.

Behavioral intentions were measured in terms of user intention to continue using the app and their intention to return, and survey respondents indicated a strong willingness (4.46 mean) to continue using a smart product like that presented in this case study.

To assess the ease of use, two key aspects were examined: ease of decision making and ease of initial learning. Ease of decision making refers to the ability of users to make choices effortlessly with 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. In this case, ease of decision making presented a positive result as evidenced by an average score of 4.08. In contrast, when evaluating the ease of initial learning, opinions exhibited greater variability, as reflected in a higher standard deviation of 1.15. While the average score of 3.67 indicates a generally favorable perception of the app instructional support for first-time users,

Interaction adequacy was assessed to evaluate user perceptions on the app effectiveness in facilitating specific interactions. Notably, users expressed a high level of satisfaction, with an average score of 4.17, regarding the ease of finding recommended services. However, when assessing the app ability to provide an appropriate means for expressing preferences, opinions were generally positive but exhibited greater variability, as reflected in the average score of 3.83 and a higher standard deviation of 1.33.

Perceived usefulness, on the other hand, refers to the extent in which users felt that using

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
	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
Transparency & Control	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

Table 5.6: Case Study 1: UX Constructs and Survey results

the recommended services would enhance their performance as a support mechanism. The high average score of 4.33 indicates a significantly positive response, suggesting that users generally feel well-supported in using the app to discover services that align with their needs. The relatively low standard deviation of 0.78 indicates a more consistent level of agreement among participants.

As for quality, one aspect evaluated was how accurately the recommendations matched user preferences. Overall, users expressed a high level of satisfaction with the app's ability to match recommended services to their preferences, as indicated by the positive average score of 4.0. Context compatibility examined whether the system understood the personal context requirements of users, and the responses indicate that the services recommended took these requirements into consideration to a satisfactory degree (average score: 3.75). Additionally, users reported a strong familiarity with the recommended services, with a high average score of 4.25. 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 (average score: 3.58). Notably, the app received lower ratings in terms of introducing users to novel services, with an average score of 2.83. The final construct evaluated was transparency & control. Control assessed whether users felt in control during their interaction with the app. Feedback on this was neutral, suggesting that users did not feel completely in control of the decision-making process (average score: 3.33). In the case of transparency, users strongly agreed that they understood why certain services were being recommended to them (average score: 4.33).

In addition to the questionnaire, an online evaluation was conducted to measure metrics relevant to the performance and effectiveness of the recommendation component (Herlocker et al., 2004).

In line with Hypothesis 1 presented in Chapter 1, the research question for this evaluation is: *Do users who engage with recommended services experience a reduction in task completion time compared to those who do not follow the recommendations?*

To address this question, the metric, "Time-on-task" was employed. The average time-on-task, measured in seconds, was assessed for users utilizing the "Recommended for you" tab and those opting for manual processes (Table 5.7). This metric allowed us to compare the time taken by users who did not utilize the recommended services with that of those who did. Notably, users engaging with the recommended services completed the task 55% faster. This serves as an indicator of the ease of use of the app.

Statistical analysis, utilizing a t-test on the 'Time-on-Task' values, provided evidence of a significant difference between the two user groups (T-Statistic: -2.957, P-Value: 0.0045). This outcome demonstrates that users who engaged with the recommended services experienced a significant reduction in task completion time compared to those who did not follow the recommendations. The negative T-Statistic further underscores this difference, indicating that, on average, users using the suggested services completed tasks faster. With a p-value less than the significance level of 0.05, there is a high degree of confidence in rejecting the null hypothesis. In this analysis, the engagement with recommended services served as the independent variable, distinguishing the two groups.

The second research question is: *To what extent are the service recommendations accepted by users?*

The CTR metric gauges the efficacy of service recommendations by quantifying the frequency of successful acceptance by users. This measure is independent of the ranking of presented services (limited to the top 3 options). As presented in Table 5.7, the CTR stands at

69.13%.

Metric	Result
$\text{CTR} = \frac{\text{Total Number of Acceptances}}{\text{Total Number of Sequences}}$	0.6913
$\text{Avg. Time-on-Task} = \frac{\text{TT\_1+TT\_2+...+TT\_n}}{n}$	00:49 seconds (accepted) 01:44 seconds (rejected)

Table 5.7: Usability metrics: Online Evaluation

5.6 Results and Discussion

This case study investigated the impact and validation of implementing AdaptUI. In this section, we discuss the results.

5.6.1 Validation of performance and usability

Analysis of performance and usability can be related to efficiency and *ease of use* (Pu et al., 2011). In the questionnaire, users strongly agreed that the *"recommended for you"* tab on the app saved them time as presented on the results of *ease of decision making* subclass (Table 5.6). This feature—the result of adaptation and personalization—is a helpful decision-making tool which results 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 usage of the system or in their sense of competence. Importantly, the results reveal a strong positive correlation (r = 0.64) between user perceived usefulness and the ease of decision-making (Figure5.8).

Ease of decision making can be considered a form of efficiency, as it reduces the cognitive load and effort required. This observation is supported by the data collected, which shows a decrease in the time spent on tasks as compared to manual execution. Statistical analysis of time-on-task revealed a significant reduction when using the recommendation component, and also highlighted the positive influence on user efficiency. Combining task completion time with user feedback provides a more comprehensive understanding of user behavior and emphasizes the importance of an interface that supports users in finding suitable services quickly and conveniently.

In terms of *ease of initial learning*, this subclass scored a lower value in the *ease of use* construct. While users perceived the adaptation of the interface as a time-saving tool, it may not have provided strong instructional support during the initial stages of use or improved the learnability of the system. This suggests that users may have required some time to familiarize themselves with the digital services provided by the app to understand how to effectively leverage its recommendations. Further attention could be paid to improving the instructional and learnability aspects, particularly during the initial stages of user interaction with the smart products and service system, when it is not immediately apparent how recommendations are generated. 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 ability of the app to quickly grasp and adapt to the individual needs of users.

Interaction adequacy also played a crucial role, as users stated that the recommended services in the app were easy to find and provided an appropriate means to express preferences. This indicates that a well-designed and intuitive interface contributes to a positive UX and enhances usability. Nonetheless, is important to note that *interaction adequacy* exhibits a strong correlation with *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 needs and context of the user.

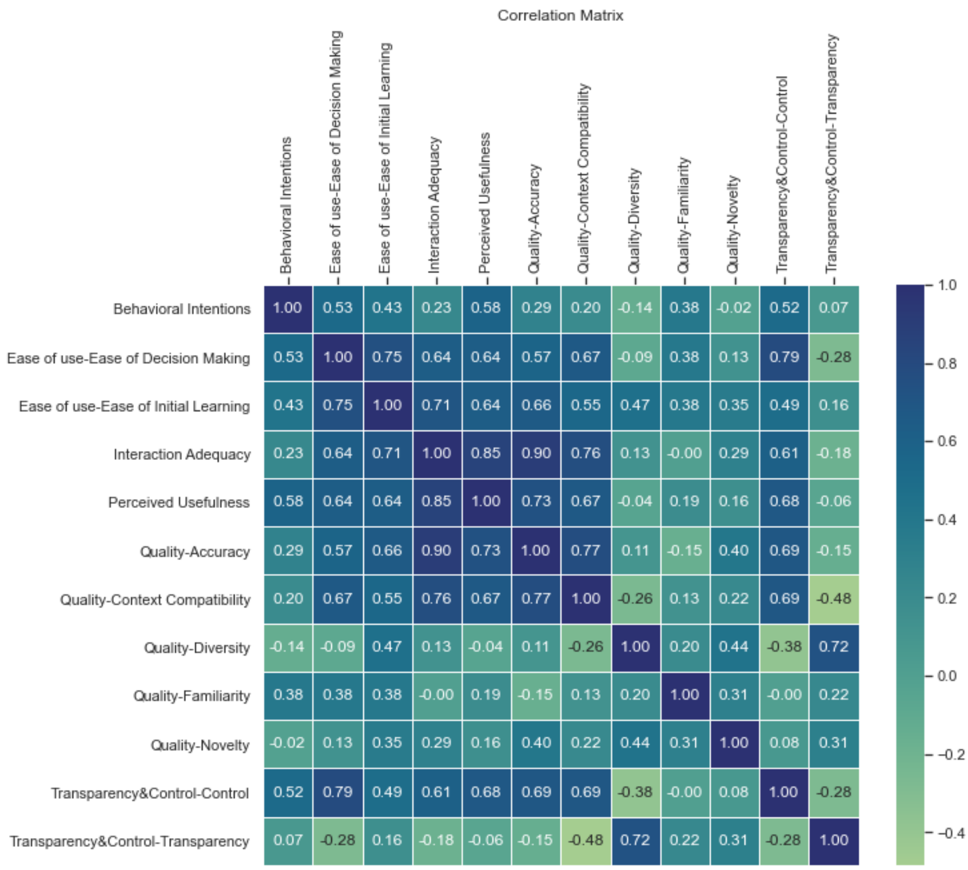


Figure 5.8: Correlation Matrix UX Constructs in RS

5.6.2 Validation of user engagement and satisfaction in S-PSS

Engagement refers to the level of user involvement with a particular entity. One commonly used metric to measure engagement is CTR, and our results show that the app recommendations were used 69.13% of the time. It is important to note, however, that CTR does not necessarily correlate with metrics in the offline evaluation (i.e. Precision@k) and can potentially increase as users interact more with the system and feed the recommendation system.

Analyzing the questionnaire results, it is evident that several factors can influence user engagement. For instance, 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* positively correlated with the *context compatibility* of services being delivered via the interface, as presented in Figure 5.8. This indicates that when the recommended services align with user preferences and consider their personal context requirements, it positively impacts their engagement. This is also supported by the CTR which shows that users favoured the recommendations-based personalized interface more than fifty percent of times.

On the other hand, user satisfaction can be influenced by many factors and should not be solely attributed to high recommendation accuracy (Kaminskas, Bridge, 2016). Our findings further support this notion: the *behavioural intentions* construct reported a high level of satisfaction and desire to return to the product (Figure 5.8). This construct had a weaker correlation with *accuracy* and *context compatibility*, which indicates that user satisfaction is closely related to ease of use (McNee et al., 2006). In this particular case, "ease of use" relates to 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, together with accurate and compatible adaptations for user interfaces.

5.6.3 Limitations

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

  In our case study, users exhibited a preference for recommendations of previously experienced items, which corresponds to the assertion of (Kaminskas, Bridge, 2016) that “familiarity" can reinforce trust in a system. Users strongly agreed that the recommended services were familiar to them. Novelty, on the other hand, assessed the extent to which users received recommendations for new types of services that they wanted to accept or test. In this category users expressed disagreement, indicating that they did not discover new services beyond their existing preferences (2.8/5 score).

  Implementation-wise, some users felt that the app did not assist them in finding anything new. It is true that users might feel frustrated by excessive familiarity and lack of novelty and diversity depending on the scenario (i.e. music, streaming services). However, in the adaptive user interfaces created in this study, there were no restrictions on suggesting services that users had already experienced in the past, which likely contributed to the sense of strong familiarity with the services recommended. One user suggested that the absence of images made it difficult to be persuaded to explore options beyond those they already knew. This highlights the importance of user interface design and visual elements. Interestingly, this lack of novelty did not impact their intention to continue using a product of this nature in the future.

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. Similarly, this perceived lack of diversity did not affect their intention to use the system. The recommendation component developed with 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 used by similar users, they did not necessarily perceive it as novel. Therefore to strike a balance between familiarity, diversity, and novelty, it is necessary to consider the inclusion of serendipitous recommendations that are unexpected but could still be relevant (Ziarani, Ravanmehr, 2021). Incorporating serendipity into the recommendation process can encourage exploration and promote less known services, and is an aspect that should be further analyzed in the generation of adaptive user interfaces and smart products.

These findings highlight the need for additional research into how user preferences for familiarity and the perceived diversity of recommendations may vary from product to product. Understanding these dynamics will provide insights for improving recommendation algorithms and thus tailor them to effectively meet user expectations and preferences.

- *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*, 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 in S-PSS.

5.7 Conclusions

To validate the practical implementation of AdaptUI, a case study was conducted with a hands-on development approach, considering technological aspects and utilizing appropriate tools. Furthermore, the framework presented a decoupled approach for the generation of AUIs, in which the creation of user interfaces is separated from specific application logic. The study examined various aspects of UX, including user performance, usability, engagement, and satisfaction. The implementation of the framework resulted in positive outcomes, improving both user efficiency and the ease of decision process. The user interface, developed through the framework, not only saved time but also enhanced user confidence, aligning with user perceptions regarding constructs related to "ease of decision making" and "perceived usefulness". As a result, the case study addresses Hypothesis 1:

> The implementation of the framework resulted in a quantifiable positive impact on the overall UX, with a 55% reduction in time-on-task for users engaging with recommended services. Statistical analysis indicated a significant decrease in task completion time for those who followed recommendations. An average score of 4.08 for ease of decision-making further supports the hypothesis of enhanced ease of use. Moreover, results on users behavioral intentions (mean 4.41) to continue using the product underscore their satisfaction with its usability.

Despite these positive effects, the ease of initial learning did not receive similarly high ratings, suggesting that users might need additional time to acquaint themselves with digital services and comprehend the underlying mechanisms guiding recommendations.

The results also demonstrate an improvement in the accuracy of smart product recommendations using a context-aware approach supporting Hypothesis 2:

> The Context-Awareness capability enhances the quality of adaptations in the UI of S-PSS during the usage stage, in which using a context-aware approach represented an overall improvement of $\approx 4\%$. Among the distance metrics evaluated, the cosine distance was the top performer, showcasing the most favorable results. Following closely, the Pearson distance also demonstrated substantial improvement, while the Euclidean distance, comparatively, was outperformed by the other metrics.

User engagement and usability were strongly influenced by the quality of recommendations, including accuracy, and context compatibility. The findings revealed that when recommended services aligned with user preferences and considered their personal context requirements, this 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 user feedback, emphasizing the importance of a streamlined user interface and a system that supports quick and convenient service discovery.

While the case study provided valuable insights, there are 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 user preferences across different smart products. Additionally, we should not overlook the fact that the UX is significantly influenced by key aspects of interface design, including readability, clarity, and visual elements, all of which have impact beyond the functionality of the interface. Nonetheless, this case study contributes to understanding the impact of adaptive user experiences in S-PSS, and highlights the significance of efficient decision-making, and personalized recommendations.

6 Case Study: Monitoring dashboards for Industrial S-PSS

6.1 Introduction

Industrial Smart Product Service Systems (S-PSS) are not new in the industrial sector. These systems—characterized by the integration of intelligent machines, sensors, and data analytics—have been redefining traditional approaches to both product functionality and ongoing service provision. This chapter presents a case study which examines the digital platform and user interface provided for a monitorization service for grinding machines in the framework of an Industrial S-PSS.

Industrial dashboards play a pivotal role as the UI for grinding machine monitorization e-services. They are real-time data visualization platforms, which collect, process, and present essential performance metrics, and offer decision-makers a comprehensive overview of the operational status of the machines. By seamlessly integrating sensor data and facilitating data-driven decision-making, dashboards have become integral components in the efficient orchestration of an Industrial S-PSS. They offer not only operational insights, but also present an attractive value proposition for both customers and users. Thus dashboards have become indispensable tools in manufacturing and industrial operations, providing insights into key parameters and playing a crucial role in enhancing productivity, reducing downtime, and ensuring quality standards.

In the subsequent subsections, the case study elements are examined. First, the industrial dashboard as case study is presented, followed by an analysis of the technical implementation of the framework. Then, the validation processes for both the recommendation component and UX is described. Finally, the conclusions section presents significant findings and the limitations of the study.

6.2 Case Study Set-up

The 'Timeseries Visualizer' dashboard, developed by IDEKO, a research center in the Basque Country, Spain, served as the platform for this case study. In contrast with the previous case study, the AdaptUI framework was integrated into an existing web application for monitoring services, which is currently an essential part of the current value proposition. The 'Timeseries Visualizer' application was created with Angular for the front-end and Flask for the back-end. It serves as the API and its data is stored in a MongoDB database. The primary objective of this application is to monitor grinding machines through the collection of multiple signals. Users can filter information based on specific time periods. The dashboard caters to various company clients, including user roles such as researchers, testers, and maintenance operators. Each of these specific roles are tailored to the requirements of each client.

The Timeseries Visualizer consists of three main views:

- Login: Users can log into the application with secure user authentication.
- Machine Selection: Users select the specific grinding machine which requires monitoring, as illustrated in Figure 6.1.
- Dashboard: Features a left-side navigation bar and presents visual representations, including line plots from the signals received from the grinding machines. The information presented in the dashboard depends on the company, machine selected, and additional filters available, as shown in Figure 6.2.

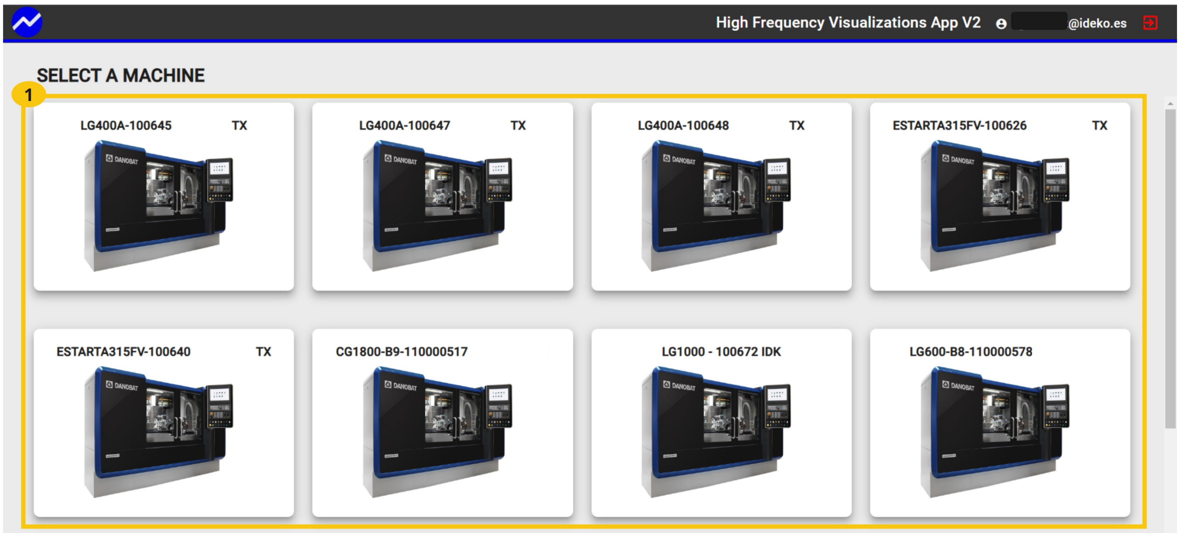


Figure 6.1: Machine selection "Time Series Visualizer"

While the application serves multiple clients, this case study specifically focused on internal users researchers from IDEKO. The interactions of 10 users were systematically recorded over

Figure 6.2: Dashboard Panel "Time Series Visualizer"

a 48-day period. (Table 6.1) Prior to participation, explicit consent was obtained from each user, confirming their willingness to have their interactions recorded within the application.

The dataset comprises a total of 6,906 interactions. All data underwent anonymization, and does not contain any sensitive user information.

Attribute	Item	Freq.	Attribute	Item	Freq.
Gender	Male	8	Occupation	NDTResearcher	2
	Female	2		GrindingResearcher	3
				GeneralResearcher	3
				PrecisionResearcher	2
Total Users:	10				

Table 6.1: Participant demographics

6.3 Analysis & Technical implementation of Data Acquisition and Modelling

The case study commenced with an analysis of both the collection and modeling stages within the framework, as these stages are highly interdependent.

The first implementation step involved identifying e-service and e-sub-service options within the user interface. These digital services can be linked to the services in the value proposition of the S-PSS. Table 6.2 presents the selected e-services for this case study, together with an example of the group of sub-service options related to each one.

It was also important to identify the contextual information required to produce the adaptations. In general, contextual data was treated as *inclusive* information for the framework and the process of contextual pre-filtering. An inclusive parameter can be defined as a parameter or set of parameters that are considered relevant and actively contribute to a process or system. Inclusion of these parameters enhances the overall functionality. Aspects related to identity and temporal context are regarded as "inclusive".

However, in this case, some contextual information needed to be treated as *exclusive*. An exclusive parameter is a type of parameter or set of parameters that serve as constraints or limitations within the process. Exclusion or restriction of this contextual data helps narrow options, or specify conditions within the context. In this case study, these parameters reduce e-subservice options, specifically contextual data related to company clients and machine specifications. This information was then uploaded into the ontology to create the corresponding individuals for the Context entity.

E-Service	Examples of E-Subservice - Options	Context
Machine Monitoring	Machine LG4004-100645 Machine LG4004-100647 etc	Company (exclusive) User Role (inclusive) Shift (inclusive)
Schema Monitoring	Schema TDOMONWH1X1 Schema TDOMONWH2Z1 etc	MachineType (exclusive) User Role (inclusive) Shift (inclusive)
Signal Monitoring	Signal Wheelhead2Power Signal CAxisSpeedActual etc	MachineSchema (exclusive) User Role (inclusive) Shift (inclusive)
Transformation Execution	Common - fft Common - negate Grinding - center etc	User Role (inclusive) Shift (inclusive)
Indicators Selection	Grinding- power-stats Grinding- force-stats etc	User Role (inclusive) Shift (inclusive)
Chart Management	Delete all Charts Manual Selection Files One-chart distribution etc	User Role (inclusive) Shift (inclusive)

Table 6.2: E-Services from Grinding Machine Monitorization Service

For the acquisition of user interaction data, mirroring the approach of the preceding case study, an auto-logging methodology was employed to capture user interactions without disrupting the normal functionality of the application. In the present study, we employed Google

Tag Manager (GTM) integrated with Google Analytics (GA4). Interactions are systematically logged and subsequently stored in a Big Query data warehouse. Table 6.3 describes the most relevant fields within the captured dataset, together with illustrative data examples.

To establish a relationship between the individuals instantiated in the ontology and their corresponding interactions, an event label is transmitted as a personalized field within Google Tag Manager. This label incorporates *{eservice_ID: esubservice_ID}* in a JSON format, thereby facilitating the correlation between individuals and their associated interactions.

Column	Datatype	Description	Example
interaction_id	Numeric	Sequential id	1
user_id	String	Unique identifier for the user	ggx45gsd
ga_session_id	Numeric	Google Analytics session id	1695633755
operating_system	String	User operating system	Windows
os_version	String	Version of the operating system	Windows 10
web_info.browser	String	User web browser	Chrome
web_info.hostname	String	Hostname of the web server	idz-dev-docker.ideko.es
date_timestamp	Timestamp	Timestamp of the interaction	2023-09-25 09:49:37.390751 UTC
event_timestamp	Timestamp	Timestamp of the event	1.69564E+15
event_name	String	Description of the event	selectMachine_LG4004-100645_click
event_date	Datetime	Date of the event	20230925
event_label	String	Additional information to match event to service *{eservice_ID: esubservice_ID}*	{"machine_monitoring": "LG4004-100645"}
page_location	String	URL of the web page	http://idz-dev-docker.ideko.es/
dayofweek	String	Day of the week when the interaction occurred	2
hour	Numeric	Hour of the day when the interaction occurred	9

Table 6.3: Raw Interaction Data for Grinding Machine Monitorization E-Services

Each row of the raw interactions signifies an event initiated by a user. It is important to note that the application can be concurrently engaged by multiple users. Hence, a preliminary data processing step is undertaken to organize these events into sequences, capturing the individual conduct of users within their respective sessions. In the pre-processing facet of the framework, the following tasks are executed:

i. Clearing duplicated events: Duplicated events identified in the raw data are eliminated based on the combination <*timestamp, event_name, session*>.

ii. Sequence generation: To formulate a sequence, two events are considered:

–The first marks the commencement of a session on GA4, representing a continuous period of user engagement with the application starting from the initial interaction and concluding after a period of inactivity or explicit user logout.

- The second event is the selection of a machine, since the data presented on the dashboard is pertinent to the chosen machine.
- A sequence concludes when any of the following events transpires: session termination, user logout, or a change in the selected machine.

iii. Grouping events by e-service: The final pre-processing step involves grouping events by e-service.

After the pre-processing from the 6,906 interactions 125 sequences are created, a summary is presented on Table 6.4. The interaction sequences together with their individual interactions are uploaded in batches into the ontology using the "Data Importer" (see Section 4.1.2) component of the framework.

Metric	**Average Value**
Sequence Length	29.80
Sequences per User	7.55
Most Used UI Elements	**Occurrences**
manualSelectionModal_click	255
confirmFiles_click	229
deleteChart_click	146
confirmParameters_click	141

Table 6.4: Summary of User Interaction Data

6.4 Analysis of the implementation of the Adaptive User Interface

In the process of adapting the user interfaces of the TimeSeries Visualizer, a correlation between the UI components and the previously defined e-services was established. As a result, the identified UI elements now function as the adaptation targets. The adaptation styles were chosen based on the existing UI elements in the user interface. Table 6.5 details the adaptation approach, setting out the selected adaptation targets and their corresponding styles.

In the case of *Machine Monitoring*, the identified target is a horizontal container that presents each machine available for monitoring as cards in no specific order. The adaptation style for this involves reordering the machines and highlighting the first machine within the recommendation selection given by the framework engine, as presented in Figure 6.3.

For *Schema Monitoring*, the identified adaptation target is a Drop-down list (Combobox). In this e-service, the select adaptation style approach is *Group Reordering*, indicating a focus on reorganizing items within the Drop-down list to optimize user interaction. Likewise, *Signal Monitoring* and *Transformation Execution* both target the Drop-down list as their adaptation

E-Service	Adaptation Target	UI Adaptation Style
Machine Monitoring	HBox > Cards	Item Reordering Highlighting
Schema Monitoring	Drop-down list	Group Reordering
Signal Monitoring	Charts & Drop-down list	Folding Reordering
Transformation Execution	Drop-down list	Group Reordering
Indicator Selection	Drop-down list	Group Reordering
Chart Management	HBox > Buttons	Folding Item Reordering

Table 6.5: Adaptation targets and styles of e-services

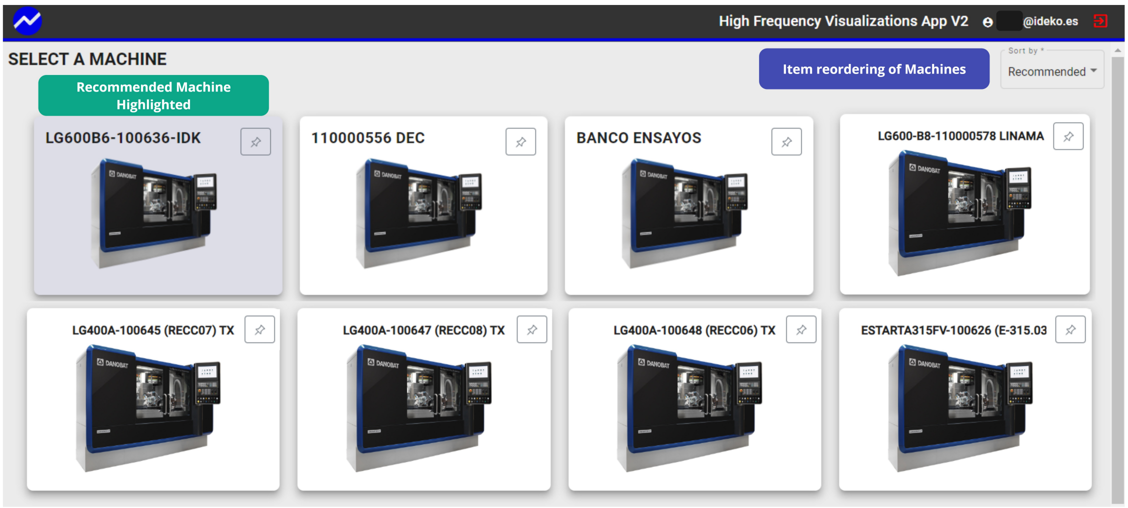
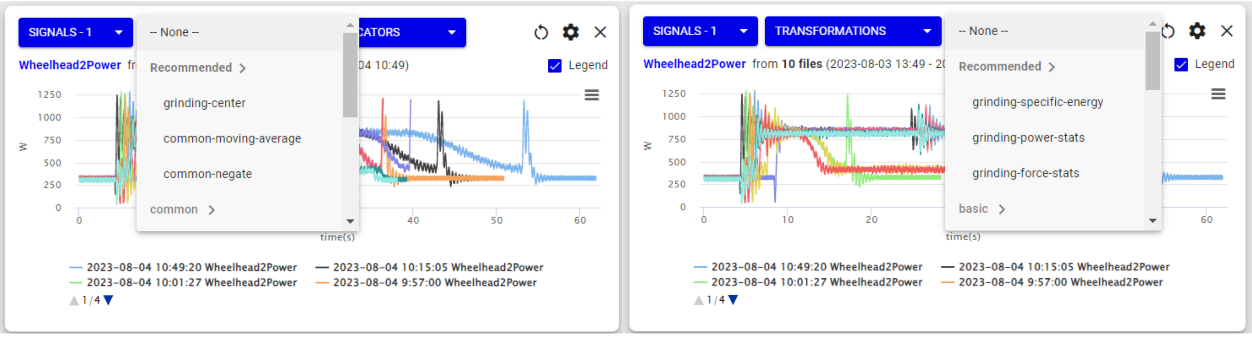


Figure 6.3: Adaptive User Interface (AUI) for Machine Selection

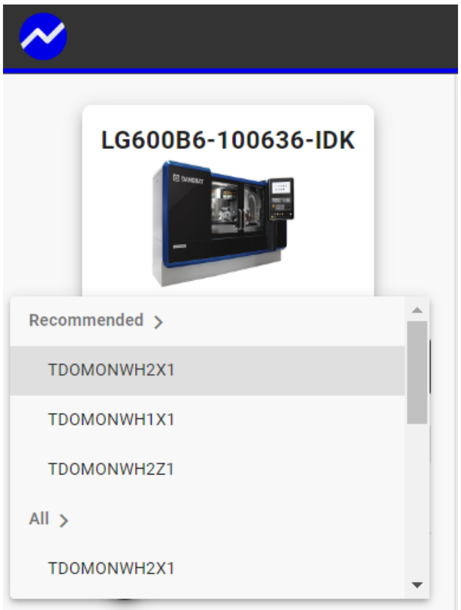
element. The chosen adaptation style is again *Group Reordering*, which delivers commonality in the approach for enhancing user experience, as presented in Figure 6.4.

In the case of *Signal Monitoring*—which involves the presentation of charts combined with combo-boxes to select signals—the dashboard presents all signals available to the user with no specific order. Using the analysis of interactions and sequence generation, it was observed that the most common tri-gram of interactions within a sequence was a set of <deleteChart*N*_click>. Thus for this adaptation style, the decision was made to fold irrelevant charts, reorder the charts and signals based on the results of the recommendation component, as presented in Figure 6.5.

Finally, for *Chart Management*, the adaptation targets are buttons that trigger the execution of corresponding actions. In this interface, a menu-based "Help" option was introduced,



(a) Transformations & Indicators



(b) Scheme.

Figure 6.4: Group reordering adaptation style: Indicator Selection, Transformation Execution, and Schema Monitoring

truncating irrelevant options and presenting only the k recommended actions. These actions are predefined call-to-actions, aiming to persuade users to utilize a pre-selection as presented in Figure 6.6.

Once the "Adaptation Targets" were analyzed, the additional dimensions for implementing the adaptive interface aligned with the overall framework description (see Table 6.6). For example, the adaptation moment occurs automatically in response to contextual changes, based on the context variables established previously in Section 6.3.

As regards the level of user control over adaptation and explicit participation, users can choose to accept or reject the recommended options. However, in contrast with the previous case study, users do not have an explicit way to rate the recommended options, since this was not considered appropriate in this industrial context. This topic, is discussed in greater detail in the following section.

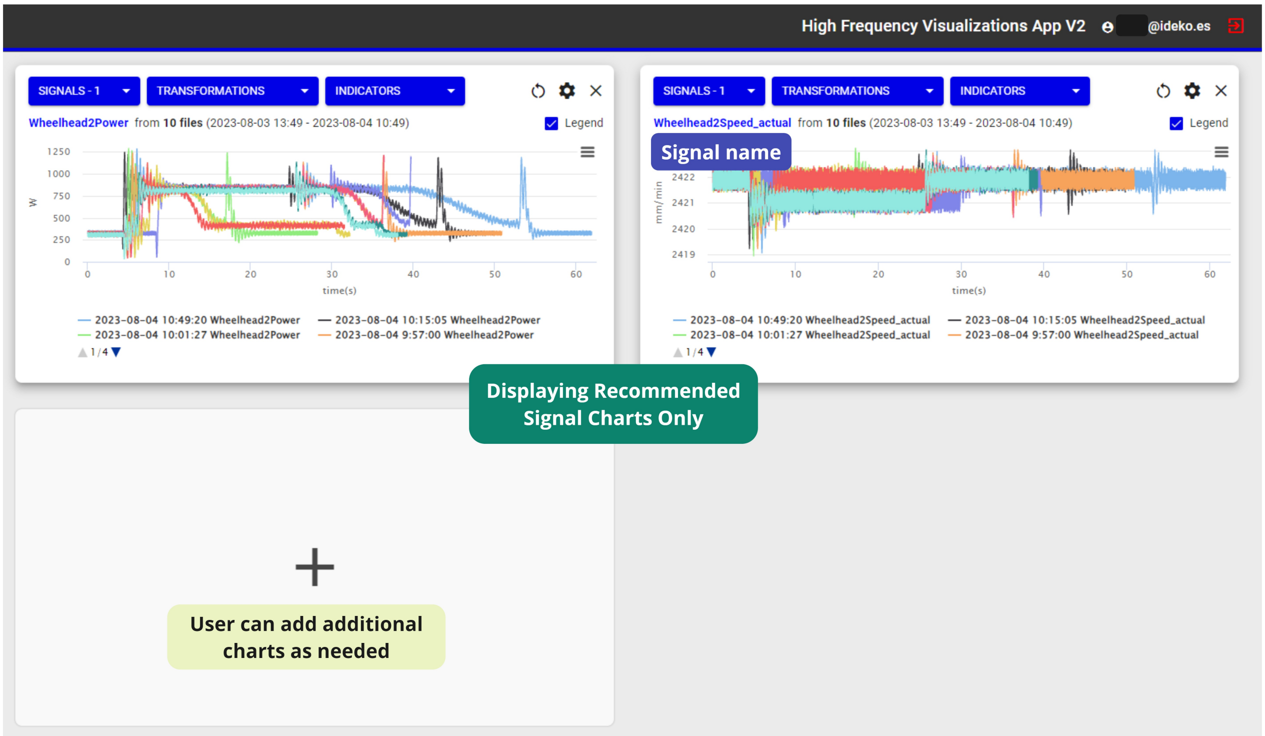


Figure 6.5: Adaptive User Interface (AUI) for Signal Monitoring service

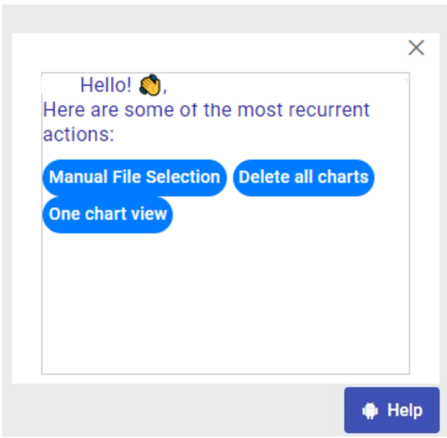


Figure 6.6: Adaptive User Interface (AUI) for Chart Management

Users can also be said to have implicit participation as their interactions with the application are captured. The visibility of adaptations is presented through the graphic elements of the interface chosen as adaptation targets. Evidently, users could be given greater participation involvement by having the ability to activate or deactivate the visualization of the adaptations and even disconnect the capture of their interactions from the app. These changes would be critical for carrying the changes of this implementation of the framework for an adaptive user interface in a productive environment for end-customers.

From the technical perspective, this is a complex user interface that demonstrates a real-world

<table>
<tr><th>Dimensions</th><th colspan="3">Application in Case Study</th></tr>
<tr><td>Adaptation Target</td><td colspan="3">TimeSeries Visualizer app</td></tr>
<tr><td>Initiator of the adaptation</td><td colspan="3">Analysis of user interactions through User-based collaborative filtering.</td></tr>
<tr><td rowspan="2">Control over the adaptation</td><td colspan="3">Recommendation-based</td></tr>
<tr><td>Automation
System identifies changes on context automatically</td><td>Participation
User can accept recommendation</td><td>Visibility
Cards with recommended services are shown to user</td></tr>
<tr><td>Moment of the adaptation</td><td colspan="3">Changes on contextual data:</td></tr>
<tr><td>Evaluation of the adaptation</td><td colspan="3">Feedback Based: Survey with participants
Analytic based: Precision</td></tr>
</table>

Table 6.6: Dimensions applied on the Dashboard Monitoring Case Study

scenario, and it would thus be difficult that all elements within the user interface adapt. For this reason, the use of templates is limited to sections of the interface where the adaptations occur. For this case study, the original model was extended to include additional UI Elements, introducing the concept of complex UI elements, through the IUIComplexElement interface. Complex elements extend the basic UI element interface and include additional fields such as options. The Option type is introduced to represent selectable options within these complex elements. Then, CheckboxDropdownList and DropdownList, implement this interface, showcasing examples of UI elements with more complex features. A view of the complete model is presented in Figure 6.7.

It is important to note that complex elements contain the attribute *isSubserviceHolder* as a boolean, and thus it can be determined if the element will hold sub-service options. As a result, sub-services will not necessarily need to have associated templates and the *template* attribute will allow null values as long as a complex element acts as a holder of sub-service options. A JSON response of the API is presented in Figure 6.8. In this case the section contains a label and dropdown list that is created from the recommended sub-service options.

6.5 Evaluation

The framework implementation is evaluated 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, derived from real-time metrics collected during usage, is presented.

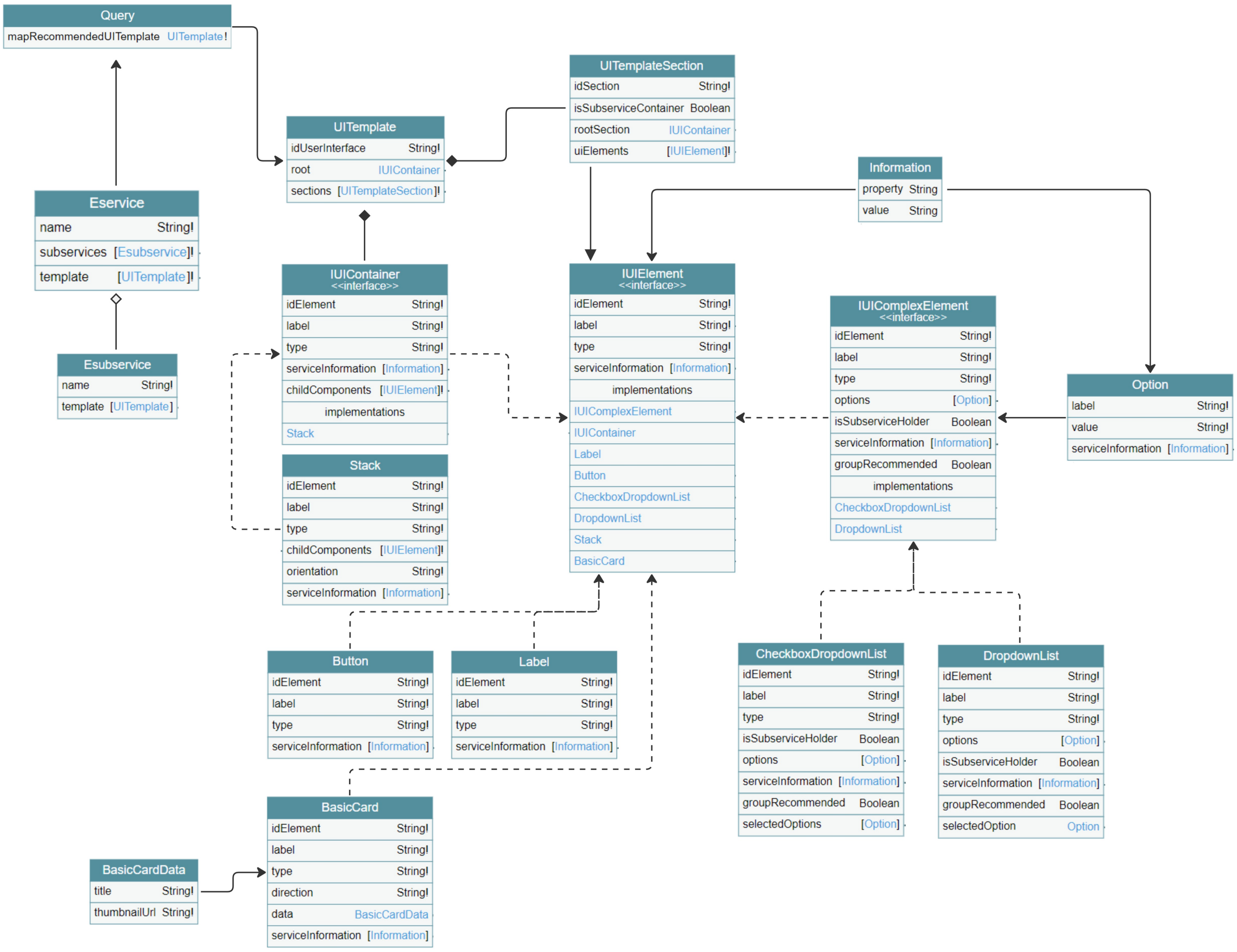


Figure 6.7: Updated data schema for UID

6.5.1 Evaluation of the Recommendation Component

Each interaction for a e-service is given a binary value, depending if the interaction represents a positive engagement (1) or negative engagement (0). To obtain the ratings, we adhere to the process described on section 4.1.3.

The offline evaluation of the present case study followed the same protocol as presented in Chapter 5.5.1 to maintain consistency. The metrics evaluated were Precision@K and MAE. Each set of experiments was repeated 20 times using randomly selected training and test sets each time (70-30 ratio respectively), then the average values were taken for each of the metrics. The primary objective was to evaluate the effectiveness of incorporating contextual information in improving recommendation accuracy. Precision scores, measured at different k values (1, 2, and 3) and utilizing various distance metrics (cosine, euclidean, and pearson), serve as the key performance indicators.

The first experiment compared the precision@k without using contextual data with a kNN model using the Pearson correlation factor, together with cosine and Euclidean distance

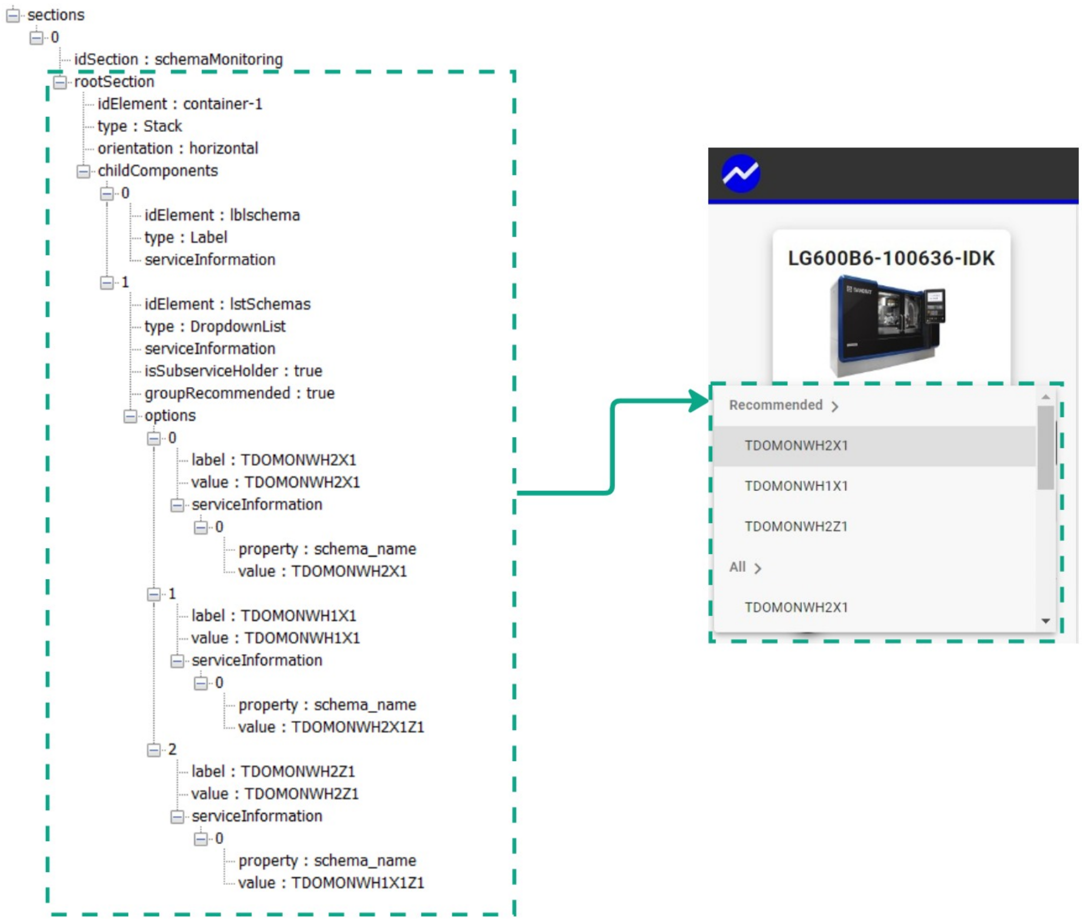


Figure 6.8: JSON response from the Schema monitoring sub-service options

metrics. The precision@K scores were evaluated at k=1, 2, and 3, as illustrated in Figure 6.9. Precision@1, ranging from 0.53 to 0.58 across a range of distance metrics, indicates that approximately 53-58% of the top recommendations were accurate. As k increases to 2 and 3, there is a marginal decline in precision values, suggesting a potential trade-off between the number of recommendations provided and their accuracy. Notably, the euclidean distance metric consistently presented the least favorable results, while the cosine and pearson distances performed similarly.

The results of the second experiment, using Precision@k with contextual data are presented in Figure 6.10, and Table 6.7. In contrast to the first experiment, the context-aware approach delivered a significant improvement in precision scores at k= 1,2. Precision@1 experienced a significant improvement, reaching approximately 0.82, which signifies considerable enhancement in the precision of the top recommendation. As k increased, precision decreased. Nevertheless, the results still represent an improvement over the no-context approach, underscoring the benefits of context-awareness.

Complementing the precision analysis, the MAE serves as a quantitative measure of the

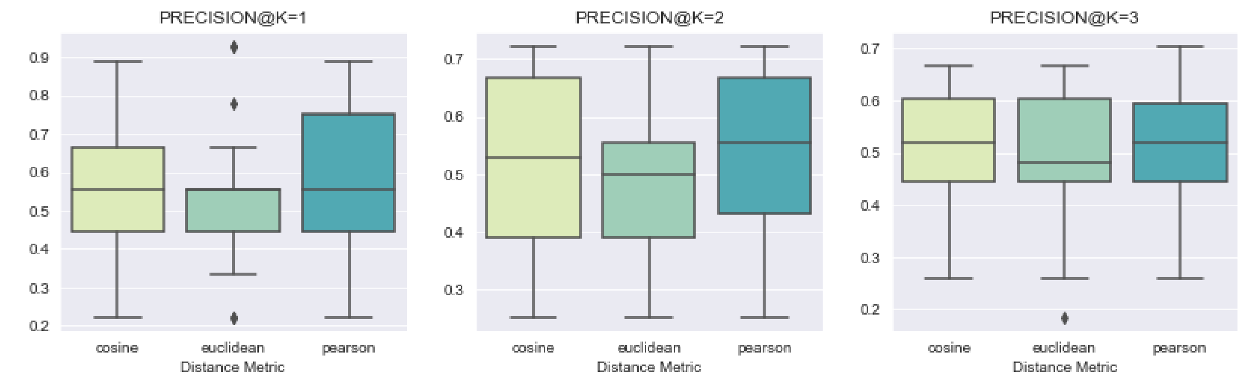


Figure 6.9: Boxplot Precision@K without context

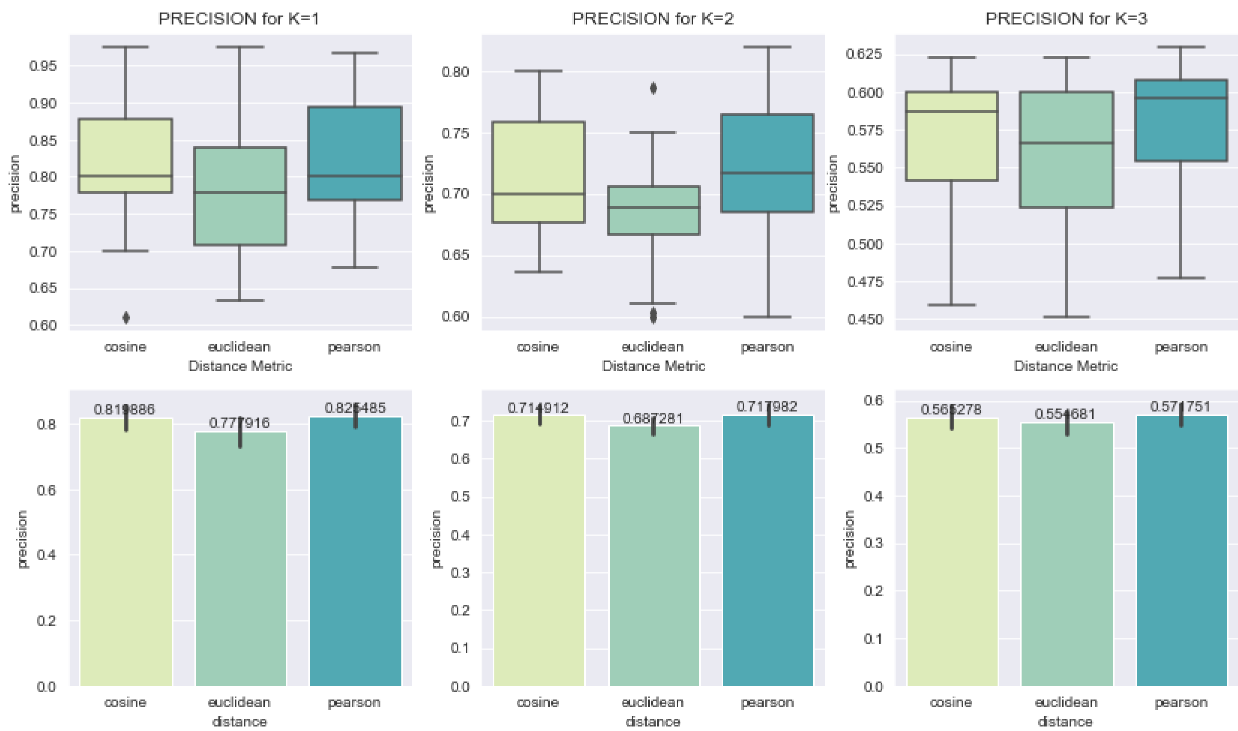


Figure 6.10: Precision@K Results with context-aware.

predictive accuracy of the recommendation system. Lower MAE values indicate more accurate predictions. In Table 6.7, the "With Contextual Data" approach reports lower MAE values than the "No Context" scenario for all three distance metrics. This reduction in MAE demonstrates the effectiveness of incorporating contextual data, as the predicted ratings align more closely with the actual ratings, leading to improved accuracy in the recommendations. The "With Contextual Data" scenario achieved the lowest MAE with the Pearson distance, although the difference with cosine was not statistically significant. However, a clear result is that Euclidean was outperformed by cosine and Pearson in both Precision and MAE. This could be due to the limitations of working with high dimensional data (Jannach et al., 2010).

	No Context			With Contextual Data		
	Cosine	Euclidean	Pearson	Cosine	Euclidean	Pearson
Precision@1	0.574	0.535	0.586	0.820	0.778	0.825
Precision@2	0.517	0.488	0.534	0.715	0.687	0.718
Precision@3	0.515	0.496	0.517	0.565	0.555	0.572
MAE	0.139	0.147	0.142	0.079	0.106	0.074

Table 6.7: Results of Precision@K and MAE

6.5.2 UX Quality evaluation

The protocol followed to assess the UX of the framework implementation is depicted in Figure 6.11. The evaluation utilized various materials, including a consent form, the locally deployed TimeSeries visualizer with UI adaptations, Tobii Stick Eye-Tracking with a Logitech Webcam for gaze recording, and screen recording software. Alongside these tools, clear instructions for general tasks were established, and the questionnaire was designed.

Participants engaged in one-to-one 30-minute sessions, and 6 out of 10 registered users participated in the evaluation. Prior to executing the planned tasks, users were briefed on the nature of the experiment and completed the consent form.

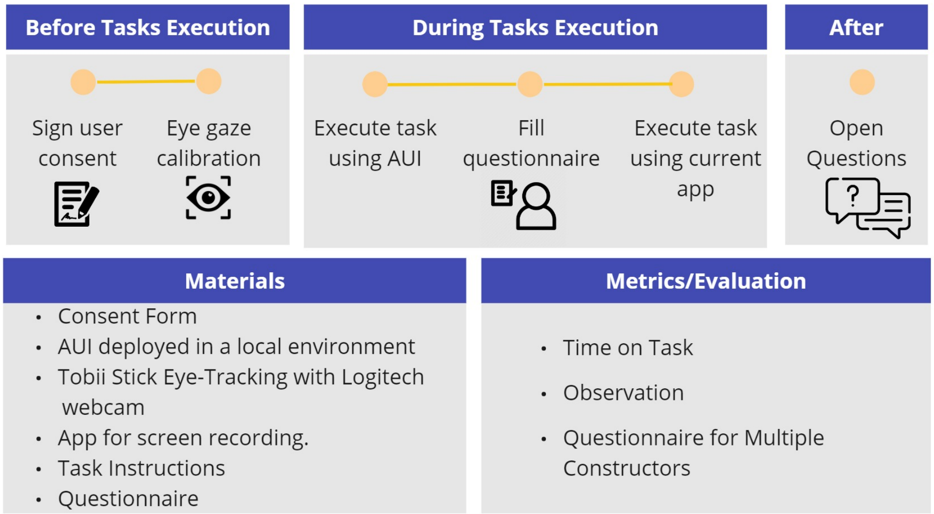


Figure 6.11: Experiment protocol followed for Case Study 2

During the session, an eye gaze calibration was performed using the Tobii Stick Eye-Tracking app. The primary goal of the eye gaze tracking was to assess whether users were able to discern the adaptation targets and perceive the adaptations made in general. The evaluation began with users performing a task and interacting with the AUI generated through the implementation of the framework on the monitoring system and deployed in a local environment (see Figure 6.12). Participants were then asked to fill out the questionnaire and evaluate their experience with the AUI. After that, they performed a control task using the current interface without adaptations. This control task, which featured slight alterations, served as a baseline for time comparison. Post-task completion, participants engaged in open-ended questions, and were asked to share observations, report issues and express any doubts they may have encountered.

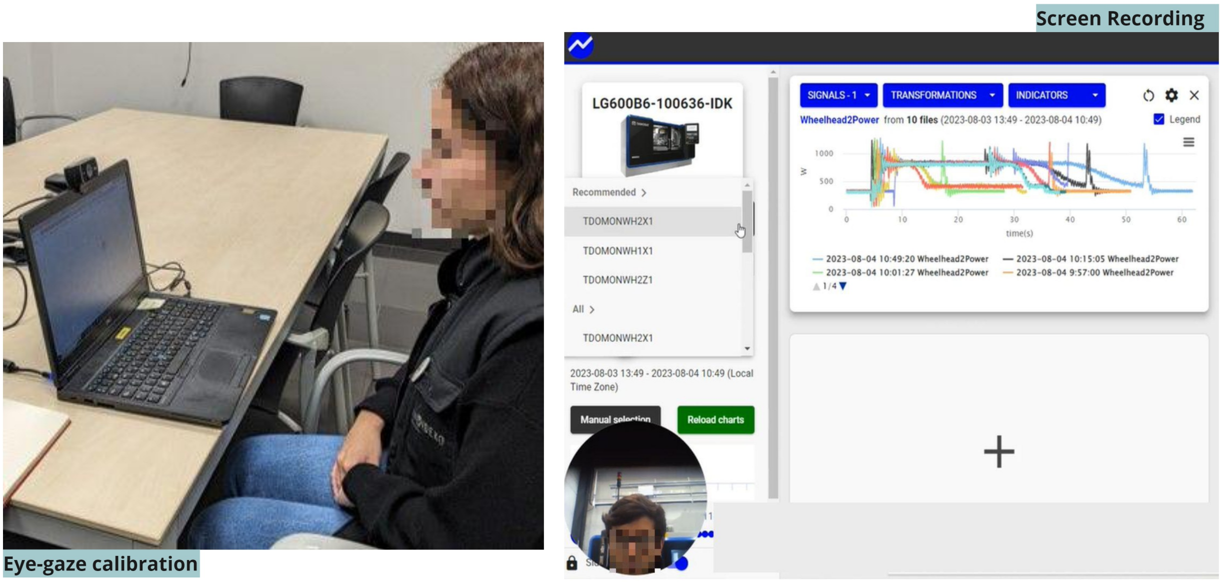


Figure 6.12: Experiment set-up: Participants and Materials

Participant	Role	App Usage	Time in Current Web App	Needed Assistance?	Time in AUI Web App	Needed Assistance?
P1	NDT Researcher	Seldom	3:11.26	2	2:39.53	1
P2	General Researcher	Seldom	1:51.07	1	2:20.83	1
P3	NDT Researcher	Regular	2:57.04	0	1:24.86	0
P4	Grinding Researcher	Regular	1:25.15	0	1:14.05	1
P5	General Researcher	Seldom	4:34.09	2	3:24.19	1
P6	Grinding Researcher	Regular	2:48.04	0	1:44.87	0

Table 6.8: Participant time evaluation

In line with Hypothesis 1, the research question for this analysis was the same as that of the first case study: *Do users who engage with recommended services experience a reduction in task completion time compared to those who do not follow the recommendations?*

The effectiveness of the AUI was evaluated by comparing user interactions with both the current user interface for the monitoring service and the proposed AUI generated through the recommendation engine. Table 6.8 presents the results of the Time-on-Task metric of the six participants (P1 to P6), each representing distinct roles and varying frequencies of usage of the application.

As stated in the protocol, the decision to start the task execution with the AUI followed by the task execution in the current application was deliberate. By initiating the session with the AUI, we aimed to capture initial user interactions without introducing a learning advantage that could potentially influence their subsequent engagement with the traditional interface. This approach seeks to mitigate the risk of users becoming overly familiar with the task, thus preserving the authenticity of the experiment. An analysis of the time spent by participants in both interfaces revealed an average reduction of approximately 39.72 seconds of the Time-on-Task when using the AUI. It is noteworthy that an increment of time using the AUI was only observed in participant 2.

$$* * *$$

$$\text{Average Reduction} = \frac{\text{Sum of Individual Reductions}}{\text{Number of Participants}}$$

$$\text{Average Reduction} = \frac{(31.73 - 29.76 + 92.18 + 11.10 + 69.90 + 63.17)}{6}$$

$$\text{Average Reduction} \approx 39.72 \text{ seconds}$$

In this case, it was possible to use a paired sample t-test because the experiment involved collecting data from the same participants using both interfaces, thereby creating paired observations for analysis. The results indicate a T-value of 2.18 and a P-value of 0.08, which suggests a discernible difference between the two interfaces, although statistical significance is not firmly established at the conventional 0.05 threshold. Hence, while a trend or impact may be associated with the AUI, further investigation and potentially a larger sample size are needed to draw more definitive conclusions.

Users also provided feedback through a questionnaire, the questions of which were developed based on the constructs previously described in Chapter 4.1.5. To capture user responses effectively, a 5-point Likert scale ranging from “strongly disagree" (1) to “strongly agree" (5) was employed for each question. The questions and results are summarized in Table 6.9 and Figure 6.13.

6.6 Results and Discussion

The present case study investigated the impact and validation of implementing a proposed framework for adaptive user experiences in S-PSS. The results are presented, and findings as well as limitations are discussed in this section.

6.6.1 Validation on performance and usability

Performance and usability are tightly correlated with efficiency and *ease of use* according to Pu et al. (2011). Ease of use can be regarded as a form of efficiency, minimizing the cognitive load and effort needed for decision-making.

In the *'Ease of use'* construct, particularly on the *'Ease of Decision Making'* subclass (Table 6.9), participants found the application more user-friendly as using the recommended options

Construct	Subclass	Question	Mean	SD
Behavioral Intentions	-	I would continue using the application with these changes that take into account my work context	4.000	0.894
Ease of use	Ease of Decision Making	Using the recommended options streamlines the use of the application	4.833	0.408
	Ease of Initial Learning	(If it were the first time using the application) I think it is intuitive and easy to navigate to find what I need	3.833	0.753
Interaction Adequacy	-	I easily found the recommended services	3.833	1.169
		I like that the application provides a suitable way to express my preferences	4.167	1.329
Perceived Usefulness	Perceived Usefulness	I feel supported in finding the options I need with the application	3.500	1.516
Quality	Accuracy	I easily found the machine I wanted to work with.	5.000	0.000
		The recommended indicators in the app are relevant to my work.	4.166	0.983
		The recommended transformations in the app are relevant to my work.	4.000	0.894
	Context Compatibility	The dashboard showed graphs relevant to my task	4.333	0.816
		The dashboard takes into account my personal context and requirements	4.167	0.408
Transparency & Control	Control	The app seems to control the decision process rather than me	3.167	1.602
	Transparency	Do you agree with the following statement? As a user, I understand why certain services are being recommended	3.667	1.033

Table 6.9: Case Study 2 - UX Constructs and Survey results

streamlines its usage. The average rating is 4.83, with a low standard deviation of 0.41, suggests a high level of agreement among participants regarding this aspect. This result finds support on the average time-on-task, that presented a reduction on task completion of $\approx$ 40 seconds (Table 6.8). This reduction represents a potential efficiency gain in user interactions, supporting the notion that the AUI could contribute positively to the overall user experience.

In contrast, the perceptions of first-time use of the application or *ease of initial learning* received a lower score than its predecessor. Participants generally considered it intuitive and easy to navigate, scoring it at an average of 3.83. However, there is a slightly higher standard deviation of 0.75, indicating some variability in participants opinions.

Interaction adequacy, within the context of usability, assesses how effectively users can engage with a system or interface to accomplish tasks successfully. An effective UI that allows users to quickly discover suitable services in their work tasks is not only achieved through the

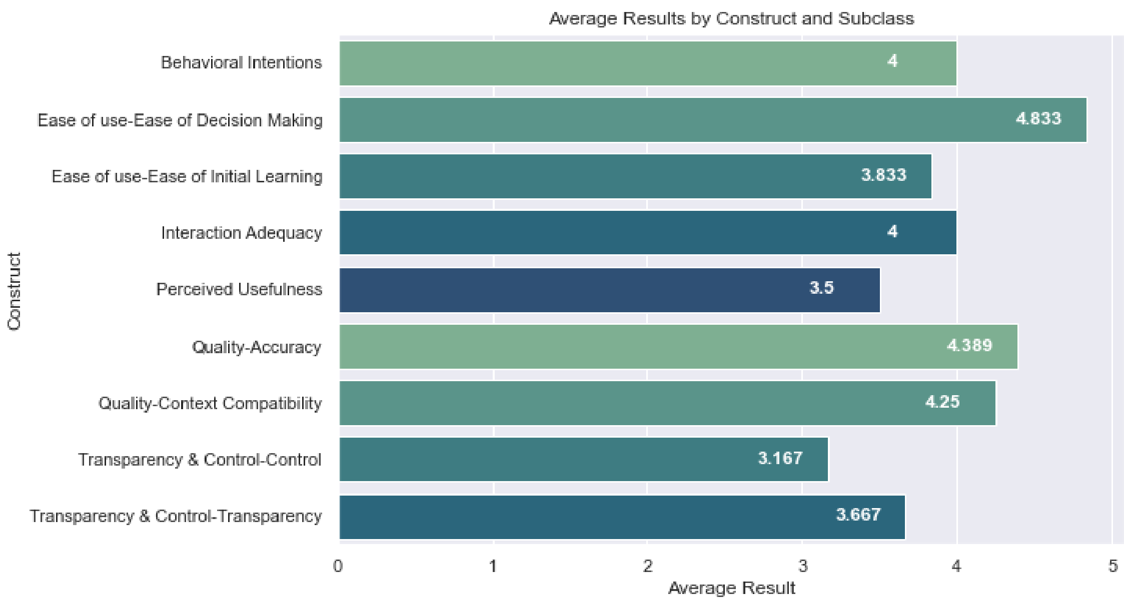


Figure 6.13: Case Study 2 - Results of UX constructs for evaluation of user perceptions

quality of recommendations but also in how well these recommendations are presented to the user.

In this case study, which explores a complex and real-world UI, adaptations are specifically tailored to sections that would benefit from real-time adjustments. Consequently, only selected sections of the UI were adapted, and templates for these modified sections were created, as detailed in Section 6.4. Various adaptation styles and targets were applied to different sections of the interface, involving element reordering, grouping, and folding.

Thus, understanding the *interaction adequacy* perception plays a crucial role that can potentially impact the perceived accuracy and usefulness of the AUI. Moderate positive feedback was received for the ability to find recommended services (Mean: 3.83), suggesting that certain presentation styles for system adaptation were more effective than others, attracting greater attention from users.

For instance, the monitoring service web-app provided actions for *chart management*. The recommendation component generated the top 3 actions, which were highlighted in the form of a button-based bot in a pop-up window. However, these failed to attract the attention of any of the participants, as the eye gaze results present on Figure 6.14. Regular users of the application continued to use the familiar options when available, and in the case of unknown options, they still did not pay attention to the pop-up window. Even infrequent users of the app failed to notice the pop-up options. These observations may help explain why the "Ease of Initial Learning" subclass of the Ease of use construct is lower (Table 6.9). This subclass is more affected by familiarization with visual elements in the interface and how easy are they to find and the complexity of the interface.

Figure 6.14 also clearly shows that the focus of the user is centered on the graphs that

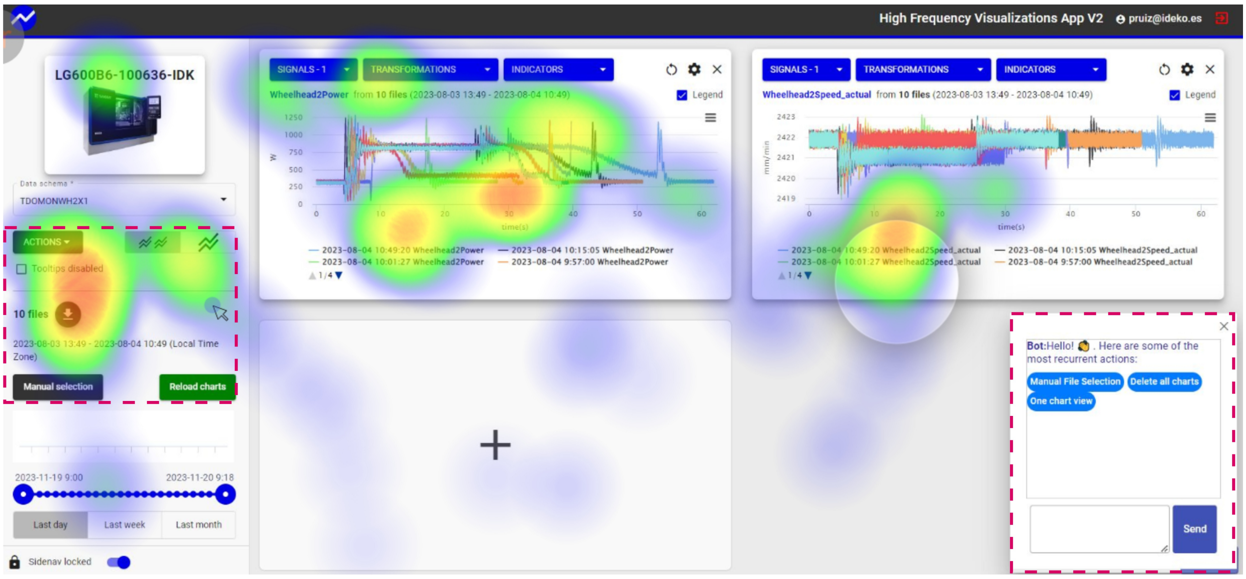


Figure 6.14: Eye gaze results from *Chart management* services

represent the recommended machine monitoring signals. In our implementation, users had the option to work with just two graphs or add more if needed, in contrast with the published version of the web-app that displayed more than 10 signal graphs. We observed that frequent users of the unaltered version of the app eliminated graphs that were not needed for them. From the audio transcription, one user mentions *"The first thing I do is eliminate all graphs"*, while another user said *"I delete many of the charts that I don't use."* Thus, folding the non-used option in the AUI was successful, and users easily found the required signals:

"The first graph, yes, for sure I use; the second one may be more for monitoring controlling forces, but it will depend on each application. For example, a user who, instead of being focused on processes, is focused on precision. Instead of these signals, he will be more interested in those of speed, position [...] because he will be more interested in analyzing the precision with which those axes move. For me, more than the precision with which they move, I am interested in seeing how it is working, how it starts; each user will have their signals for their work."

The grouping of recommended services was also easy to identify by users, as illustrated in Figure 6.15. To quote one user: *"We have a lot of data schemes, and since the names are so similar, you have to figure out which is which; grouping them by my usage makes access easier."* While other user agreed to extend this methods to other sections that were not selected for adaptation in this case study: *"Several indicators have the same variables but in different orders, there is no specific order; I would group them in some way."*

Even when a more extensive analysis of the dimensions of the visual elements on the interface is beyond the scope of the framework, it highlights that the use of AUIs and AI technologies does not replace the need for a thorough understanding, analysis of user requirements, and the importance of the design of the user interface. Still, the framework serves as a toolbox that allows for considering all design implications since the choice of adaptation target and styles

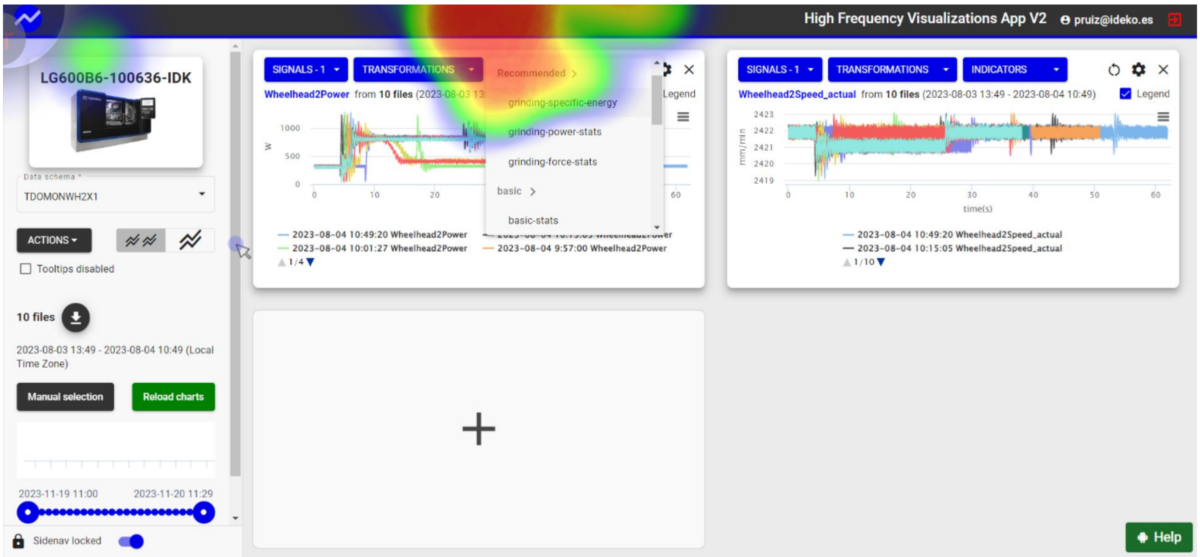


Figure 6.15: Eye gaze results from *Indicators*

will affect interaction adequacy.

6.6.2 Quality analysis and User satisfaction

The services proposed by the recommendation engine correspond as one of the main features of the AdaptUI framework. The Quality construct measures the perception of accuracy of these recommendation and if they fit user context.

In the "Accuracy" subclass users unanimously reported a high level of satisfaction (mean 5.0), indicating that they easily found the machine they wanted to work with. This unanimity suggests a user-friendly and efficient interface for accessing specific machines within the application.

Regarding the relevance of recommended features, users generally found the app suggested indicators to be pertinent to their work, scoring an average of 4.17, with some variability in responses (standard deviation: 0.98). For recommended transformations, users, on average, rated their relevance at 4.0, indicating a moderate level of agreement, with a standard deviation of 0.89.

In terms of "Context Compatibility," users found the dashboard to be effective in displaying graphs relevant to their tasks, with a mean rating of 4.33. This suggests that the visual representation of data aligns well with users work requirements. Furthermore, the dashboard consideration of users personal context and requirements received a mean rating of 4.17, indicating that users felt their individual needs were taken into account during their interactions with the application. As the results of the offline evaluation demonstrate (see Section 6.5.1), there is an improvement in the precision and performance of the recommendation engine when using a context-aware approach (Table 6.7), supporting user perceptions.

Regarding user satisfaction, assessing the willingness of users to continue using the presented smart service indicates that there is a relationship between satisfaction level and the likelihood of returning to the product. The average score of this case study reflects a high level of agreement (mean 4.00).

6.6.3 Findings and Limitations

The findings of this case study, similar to the previous one, indicate that users expressed the need for the web app to remember the parameter information required for the service to function. The monitoring of grinding machines for each selected indicator required filling in multiple parameters to align with user needs. One user expressed, regarding the indicator monitoring service: *"It would be an improvement for the application if the information were reloaded based on previous actions. Most likely, I will need to retrieve this indicator again in 10 minutes and have to fill everything out again."*. This recurring feedback emphasizes the importance of enhancing the information retention of services.

While this case study provides valuable insights, it is not without its limitations. Given that the context of this application is targeted towards an industrial work environment novelty and diversity factors were not measured in this study. This was based on the results of the previous case study (in which these factors were assessed) that indicated that a lack of diversity and novelty did not significantly impact behavioral intentions (see Section 5.6.3). In line with the previous approach, we did not filter out previously used services when generating recommendations.

Additionally, in this case study, it was not possible to include measure explicit feedback through the app, since it was a work environment, including rating options using emojis like in previous case study was not considered. Instead, we explored this option through the questionnaire where users agreed to have ways to express their preferences (mean: 4.16), something that have to further explored in future work.

Another limitation is the sample of 6 participants used for evaluation, mostly due to availability constraints, that may be considered small. However, literature suggests that only 5 users are needed to identify 80% of usability problems (Jakob, 2000). Despite ongoing discussions on this claim, it has prompted further research to test various user samples. Subsequent studies have indicated that usability tests with 6 to 8 participants can be as effective as tests with 12-15 users, allowing for the identification of both minor and major issues (Lindgaard, Chattratichart, 2007; Alroobaea, Mayhew, 2014).

In light of these considerations, we believe the case study remains relevant for its intended purpose. It builds upon an existing web app with which users had a certain degree of familiarity. Specifically, it focuses on the evaluation of adaptations and the interaction effectiveness of the web app in a well-defined task. The user group for this service shares the same profile, making the results valuable for gathering perceptions within that particular profile. Furthermore, the case study demonstrated the possibilities of implementation in a real-world scenario, focusing on different aspects of the UI.

6.7 Conclusions

This chapter presented a case study showcasing the practical implementation and validation of the AdaptUI framework in an industrial setting. Our study adopts a hands-on development approach, emphasizing both technological aspects and design dimensions crucial for effective adaptations. Notably, we successfully deployed the framework into an operational monitoring service web-app for grinding machines.

Specific sections within the digital service app were strategically chosen as adaptation targets. These adaptations were implemented in a controlled local environment, allowing us to conduct user testing and gather valuable insights into user perceptions of the UX with these adaptations.Through a thorough evaluation and discussion of the results, it was evidenced that the integration of AdaptUI streamlined the application usage. These results support the Hipothesis 1

> The implementation of the framework resulted in a quantifiable positive impact on overall UX. An average score of 4.8, coupled with a low standard deviation of 0.41 for ease of decision-making, robustly supports the hypothesis of enhanced ease of use from users perception. Furthermore, users expressed strong behavioral intentions (mean 4.0) to continue using the product, affirming their satisfaction with its usability. There was an average reduction of approximately 39 seconds in time-on-task for users utilizing the UI with adaptations.

While there is an improvement on the time-on-task, it is important to acknowledge that statistical significance was not achieved in this instance. A more extensive sample size could potentially solidify this result.

During the case study, it was incorporated an automatic logging tool to capture user interactions using Google Tag Manager and Google Analytics to capture user interactions over a 48-day period. This data enabled us to train the recommendation engine, assess the precision of the model, and evaluate the benefits of employing a context-aware approach compared to a non-context-aware one, supporting Hypothesis 2: .

> The Context-Awareness capability enhances the quality of adaptations in the UI of S-PSS during the usage stage, where employing a context-aware approach resulted in an overall improvement of the precision in approximately 14% in the offline evaluation (Section 6.5.1). Furthermore, results from UX Quality evaluation (Table 6.9) , user perception of the "Context Compatibility" construct solidify this assertion, with an average result of 4.25.

In the offline evaluation, among the evaluated distance metrics, the Pearson distance showcased the most favorable results. Following closely, the cosine distance also demonstrated a similar improvement. However, the Euclidean distance was outperformed by the other metrics. This observed behavior aligns with the results of the previous case study (see Section 5.7), indicating a consistent pattern across different contexts.

The implementation of a context-aware approach is determinant factor in achieving improvements in the quality of recommendations, underscoring the framework effectiveness in real-world industrial scenarios.

7

Conclusions

This chapter summarizes the main contributions and conclusions of this doctoral work and outlines the research directions that can be explored in future work.

7.1 Summary of Contributions

This section provides a concise overview of the contributions made in the course of this research, focusing on the development of a framework for integrating AUI into the design and utilization phases of S-PSS.

1. Preliminary work:

   In Chapter 3, a SLR was conducted, enabling the identification of emerging trends and gaps in the design of S-PSS. Subsequently, an exploration of the characteristics and elements integral to the design of S-PSS was undertaken based on the identified state-of-the-art. This examination revealed three key characteristics: data-driven value co-creation, closed-loop design, and context-awareness.

2. AdaptUI Framework Development:

   In alignment with the characteristics of S-PSS illustrated in Figure 7.1, AdaptUI was conceived to facilitate the development of AUI aligned with S-PSS characteristics. The framework not only aims to establish methodological development guidelines for generating AUIs tailored to S-PSS digital services but also addresses the dynamic nature of S-PSS design challenges. This involves the customization of interfaces to dynamically respond to user behaviors and contextual factors.

   A detailed overview of the framework is presented in Chapter 4. A significant contribution of the framework is the exploitation of internal and readily available data sources, such as user interactions in the usage stage of S-PSS, in combination with other contextual data sources. The framework core uses ontologies and data schemas to model the various actors

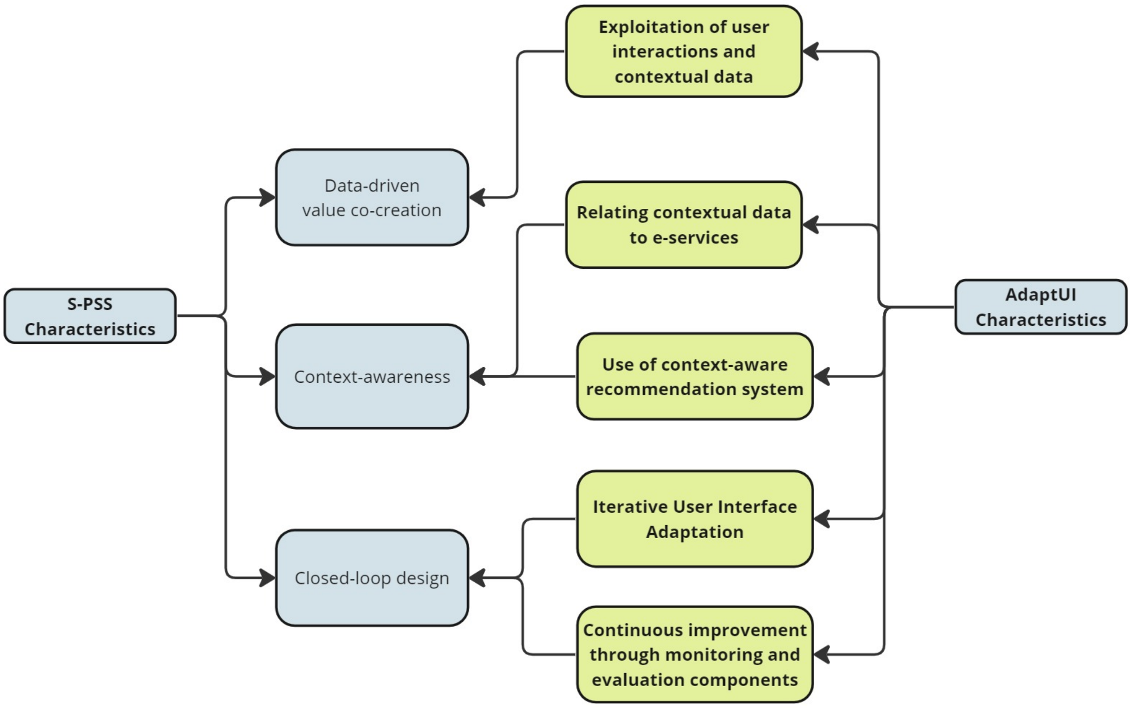


Figure 7.1: Characteristics of S-PSS aligned with AdaptUI characteristics

that intervene in S-PSS and the generation of the AUI, together with context-aware recommendation systems to generate recommendations that translate into visual elements in a UI.

3. Case Studies:

   Practical validation was demonstrated through the implementation of the framework in two distinct case studies. Figure 7.2 presents the software architecture implemented in AdaptUI, which shows that the digital platforms provided by the S-PSS are external entities to the framework, and the components of the framework were reused across the case studies which have been detailed on Section 4.2.1.

4. Technical Aspects and Design Evaluation:

   In addition to detailing the technical aspects of the implementation, the research provided an overview into the design and evaluation processes. This approach facilitated the creation of a roadmap, integrating both technical aspects and design dimensions into a unified matrix resource (refer to Section 4.2.2).

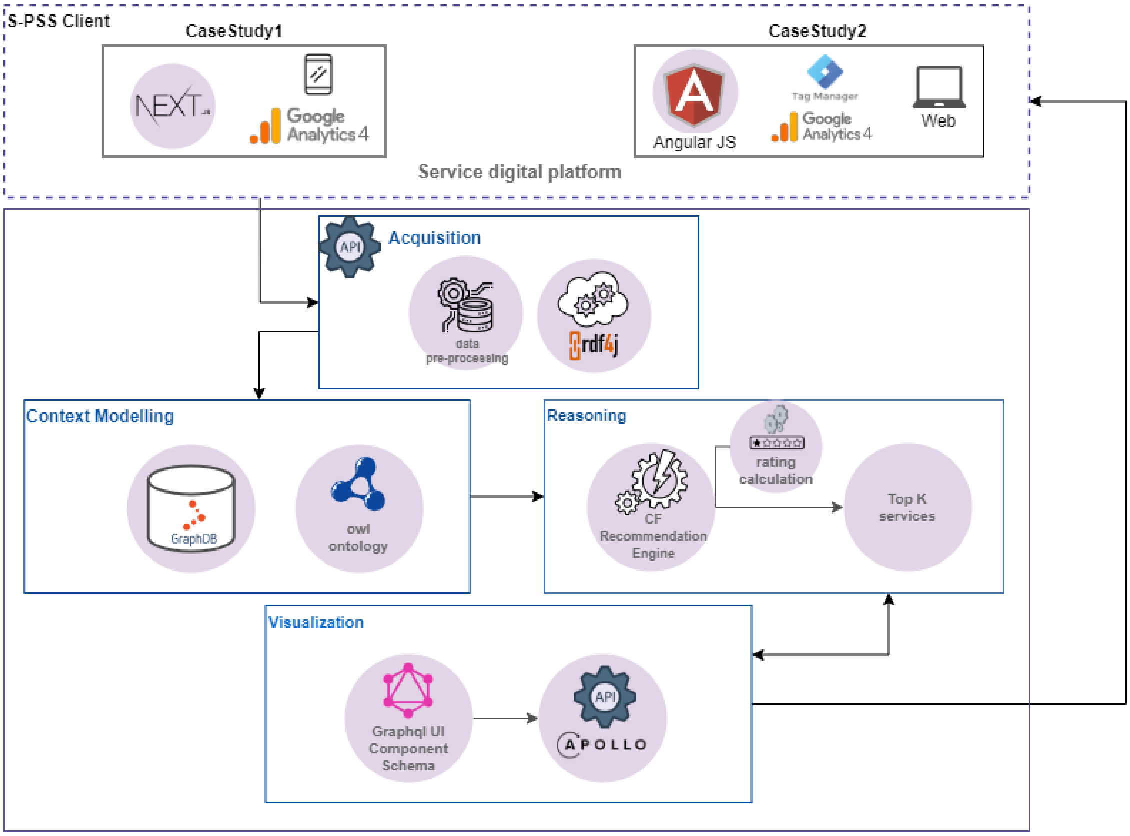


Figure 7.2: Architecture of the AdaptUI Implementation

7.2 Hypotheses validation

H1: The proposed framework results in a quantifiable positive impact on user experience, demonstrated by a favorable user perception of ease of use and overall satisfaction.

AdaptUI was applied to two case studies: (i) an app for a smart vending machine created for end-users (C1), and (ii) a web app created to monitor grinding machines in an industrial context (C2). The framework implementation was evaluated in both case studies with a questionnaire which provided valuable insight into user perceptions of the constructs employed to measure UX quality. A comparison of the responses from C1 and C2 is presented in Figure 7.3. The results of each of the usability constructs (i.e., Behavioural Intentions, Ease of use, and Quality) are plotted against a Likert scale ranging from 1 (indicating strong disagreement) to 5 (indicating strong agreement).

For the *Behavioral intentions* construct, users in both case studies responded positively to adopting the adaptive user interfaces. C1 scored an average rating of 4.42, while C2 received a lower but still positive rating of 4.00. These figures indicate a general acceptance and willingness from users to engage with the tested systems.

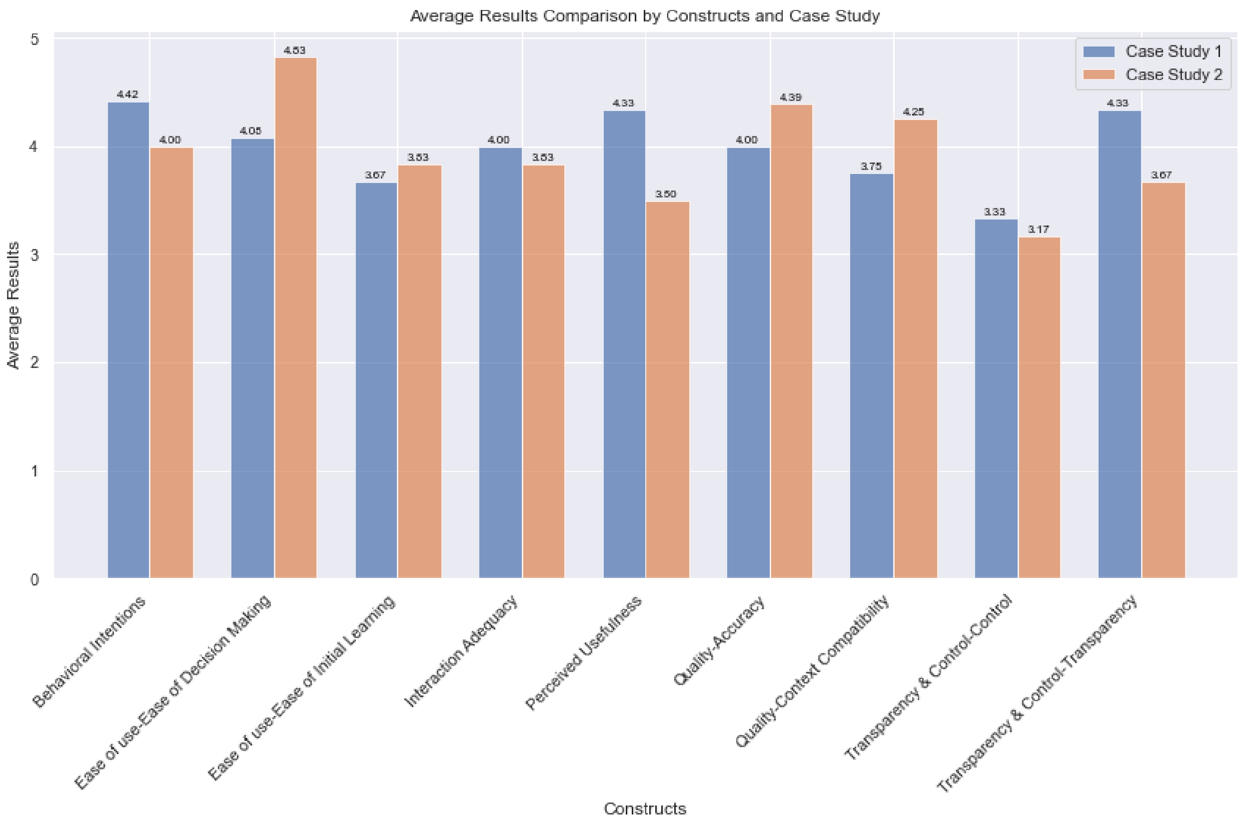


Figure 7.3: Comparison of Average Results by Constructs and Case Study

In the *Ease of use* construct, users provided feedback on two sub-classes: *ease of decision-making* and *initial learning*. C1 users rated *ease of decision-making* very favourably (4.08), and the C2 interface ranked even higher for the same sub-construct (4.83). In the case of *initial learning*, however, the results indicated that the grinding machine monitoring system could be slightly more intuitive for initial learning (3.83), than the smart vending machine (3.67).

Both studies received positive feedback for *interaction adequacy* and respondents found the applications effective in delivering recommended services. C1 reported a higher average rating (4.00) than C2 (3.83), which suggests that users of the vending machine interface found it slightly more accommodating to their preferences and needs. Furthermore, in C1 users were able to explicitly express their preferences , a feature not made available on the C2 interface. However, the second study included more adaptations which represented a more complex UI.

The *perceived usefulness* construct evaluates the extent to which users feel that the app helps them to find relevant options. While respondents from both studies ranked this construct positively, the results from C1 (4.33) were higher than those of C2 (3.50). This could be attributed to the nature of the tasks involved—i.e., users interacting with the grinding machine monitoring services may perceive a greater need for assistance in accomplishing their tasks. Therefore, any future adaptation of industrial user interfaces for S-PSS would require careful analysis of the adaptation targets.

Figure 7.4 plots an overall rating for each construct. While the case studies represent different scenarios, the figure aims to offer a general view of the UX for each construct. The validation of Hypothesis 1 is based on quantifiable impact, which can be obtained from the user perception results. On the whole, these demonstrate positive outcomes, particularly in the *ease of decision making* subclass of *ease of use*.

Additional support for hypothesis validation comes from the "*average time-on-task*" metric which reported a reduction in task duration in both case studies. In the case of C1, a decrease in time was clearly observed, confirming a significant difference. However, for C2, the evidence of a significant difference could not be established, primarily due to sample size limitations ($p = 0.08$). While the specific impact on task duration in C2 may not be statistically significant, the favorable trends in other aspects emphasize a quantifiable positive influence on UX.

Thus, Hypothesis 1 is confirmed.

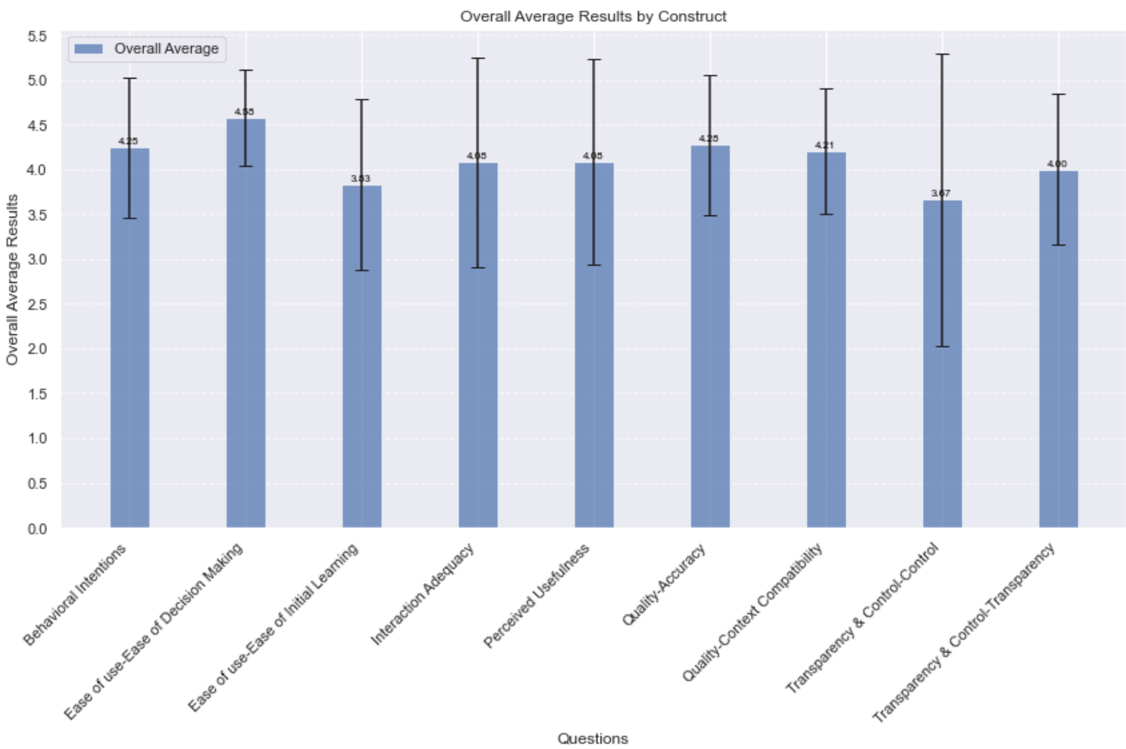


Figure 7.4: Comparison of Overall Average Results by Constructs

H2: The Context-Awareness capability enhances the quality of adaptations in the user interface of Smart Product-Service Systems during the *usage stage*, specifically in terms of accuracy, precision, and context compatibility of the adaptations.

From users perception, in the *quality* construct, both the sub-classes *accuracy* and *context compatibility* scored favourably in both case studies (Figure 7.3). C2 respondents ranked *accuracy* (4.38) and *context compatibility* (4.33) higher than C1 (4.00 and 3.75, respectively). This indicates that users of the grinding machine monitoring services perceived the recommendations as well-suited to their specific work needs. Importantly, there is an increased perception of how well the recommendations fit the context in C2 (*context compatibility*).

This trend is in line with the offline evaluation, which measured precision and MAE, (Table 7.1). The higher C2 scores could be due to the addition of exclusive contextual parameters that were needed to constraint the interactions between the machine and service context. These, in conjunction with the non-exclusive contextual data, were used to calculate the ratings when certain contextual features were fulfilled (in this case contextual features relevant to the user identity and time of the task were employed as non-exclusive context). In C2, precision and MAE also improved significantly when the context aware approach was applied. This is likely due to the diverse set of options available in this case, which better highlight the benefits of the context-aware approach. It can thus be concluded that the use of contextual information contributed to an enhancement in a diverse environment.

	C1 (No Context)			C2 (No Context)		
	Cosine	Euclidean	Pearson	Cosine	Euclidean	Pearson
Precision@1	0.665	0.689	0.600	0.574	0.535	0.586
Precision@2	0.583	0.529	0.573	0.517	0.488	0.534
Precision@3	0.527	0.492	0.556	0.515	0.496	0.517
MAE	0.114	0.133	0.102	0.139	0.147	0.142
	C1 (Context-aware)			C2 (Context-aware)		
	Cosine	Euclidean	Pearson	Cosine	Euclidean	Pearson
Precision@1	0.700	0.702	0.660	0.820	0.778	0.825
Precision@2	0.622	0.575	0.628	0.715	0.687	0.718
Precision@3	0.556	0.517	0.587	0.565	0.555	0.572
MAE	0.110	0.123	0.094	0.079	0.106	0.074

Table 7.1: Combined Results of Precision@K and MAE for C1 and C2

To validate our hypothesis, the results demonstrate a significant improvement when employing a context-aware approach with user interactions as the data source that lead to adaptations in the user interface. The application of a t-test to compare the MAE results between the two groups resulted in a p-value of 0.0402, leading us to reject the null hypothesis. This statistical significance emphasizes the impact of adopting a context-aware strategy. Hypothesis 2 is therefore confirmed.

7.3 Scope and Limitations

This study is not without its limitations, which could influence the interpretation and generalization of the findings.

The main limitation is the limited number of case studies used to validate the AdaptUI framework. Although the two conducted case studies have significantly contributed to understanding each component of the framework, it is important to acknowledge that the inclusion of a greater number would have helped refine the framework. A number of other constraints have been identified, as follows:

- Physical aspects of the S-PSS: In developing the framework in the context of S-PSS, we have considered entities relevant to the product that could accurately represent “smart" capabilities. As the framework focuses primarily on aspects of user interaction with the smart device, it is thus limited to the digital services and platforms designed for users to interact with smart products. As a result, our study is constrained to the digital service, and the user experience has not been evaluated in terms of the physical aspects of the product.

- Scope of the framework: The context of the AUI is defined by the diverse purposes it can serve. The primary focus of the framework lies in generating adaptations related to the presentation of elements and facilitating access to e-services and e-subservices offered by the S-PSS. It is therefore important to note that the framework does not extend its scope to provide instructional support in scenarios requiring step-by-step guidance. The emphasis is on enhancing the ease of use of the user interface and streamlining access to digital services, rather than offering detailed instructional assistance.

- Framework implementation: The case studies showcased a practical application of the framework. In C1, all components were deployed in a productive environment for user testing. In contrast, C2 repurposed a real-world web application currently in use by clients. As a result, the modifications made to the web application to implement the AUI were not deployed in a production environment, but rather a local development environment. Nonetheless, we were able to track user interactions using Google tag manager and Google analytics, and employ the APIs created to generate the results in Angular JS frontend.

- Implementation of the AUI: The model presented in this work successfully associates UI components with digital services and features of the S-PSS. Additionally, it employs a lightweight data interchange format like JSON and a server-driven UI approach, offering benefits such as dynamic content updates and personalization. However, this approach may extend software development times due to the need for robust server-side logic and carefully considered UI-service dependencies. It is essential to strike a balance between the advantages of adaptability and the potential impact on development timelines.

- UX evaluation: the sample size of participants, especially in C2, is relatively small. Jakob (2000), however, claims that only five users are needed to find 80% of usability issues is discussed in the field. Other studies have found that usability tests with a total of 6 to 8 participants can be as effective as tests with 12-15 users. This range has been found to be useful in identifying both minor and major issues in usability assessments (Lindgaard, Chattratichart, 2007; Alroobaea, Mayhew, 2014). Nonetheless, the effectiveness a small sample size may vary based on the context and market of the S-PSS. In cases where the product-service is targeted at a diverse range of user profiles, a larger user sample may be necessary to fully capture the nuances of the user experience.

7.4 Future work and opportunities for research

This doctoral work has identified new areas of study for future investigation. Further research might usefully explore the following:

Optimization of the Recommendation Component:

- 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 user preferences in digital products. In the conducted experiments, a well-known machine learning algorithm was employed (kNN), which has been tested and analyzed in multiple scenarios. Nonetheless, it is important to understand when to apply more robust or resource-consuming machine learning algorithms, and weigh up the trade-offs between algorithmic performance and user requirements. Although 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 could focus on specific scenarios where the benefits of employing resource-intensive algorithms outweigh these trade-offs.

- In both case studies, users expressed a need for the integration of the recommendation component into the service parametrization, meaning the information that is required for the service to work. In C1, this integration was achieved using a naive approach that simply presented the most recently used parameters in a beverage. Users reported that the application did not learn their specific service preferences, such as the preferred amount of sugar in particular beverages. Controlling this aspect could be more complex in other scenarios. For example, in C2 where the indicators required extensive data input, users expressed the need for the system to remember "at least" the last used parameters. This represents a "naive approach". Nonetheless, the effectiveness and potential for generating errors, especially in services requiring high levels of accuracy, remain open questions that should be considered in future work, including as part of the adaptation process.

- The selection of contextual data has proven to be a fundamental aspect of the framework. While the framework highlights the "generalization" and "categorization" of contextual data (See section 4.1.2), the integration of a well-defined process for the selection of contextual features can significantly affect the results of the recommendation component. In the present work, the contextual data was selected by analysing the user interface where the adaptations were made. However, machine learning techniques could be employed to better understand the features that segment the interactions and the selection of services. Yuan et al. (2023) introduced an approach for the context generalization of user-exercise modality based on sensor data in a sport-related S-PSS. This context could be utilized to establish connections between contextual features and services using a rule-based approach, facilitating recommendations and adaptations. However, there is still a need to define the process of determining the contextual features required for adaptation and to explore how this process can be generalized.

Adaptive User Interface (AUI) Development:

- The data schema presented in this document (See section 4.1.4) provides opportunities to develop methods that facilitate the creation of AUI in a more intuitive manner. Research focusing on designers, particularly service and product designers, could explore ways to create natural and user-friendly interfaces. Enhancing tools and frameworks to support designers in intuitively creating adaptive interfaces has the potential to advance user interface development practices.

- Natural user interfaces and conversational recommendation systems have witnessed remarkable growth in recent years. This trend highlights the importance of considering multi-modal interfaces within the S-PSS environment which can be adapted to user needs. A help button was included in C2 which was not noticed by users. Consequently, exploring the interaction adequacy of these interfaces, investigating how users from both industrial and commercial S-PSS respond, and understanding the cases in which they are likely to engage could be subjects for further research. Moreover, examining the broader implications of these changes for the definition of AUI is essential for comprehensive understanding.

7.5 Final Remarks

This thesis was part of the DimanD programme (Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant No. 814078). Under the umbrella of DimanD, the goal of this research project is to improve the UX of various stakeholders, by delivering the right information to the appropriate users. As more industries migrate towards servitization for more ecologically, economically and socially sustainable solutions, human needs must be considered. In this context, S-PSS emerges as a way to provide a market offer combining both product and service while maintaining a lasting relationship between provider and customer. Context-awareness is one of the more important characteristics of S-PSS, however some phases of the S-PSS lifecycle remain have received limited attention in the research literature.

A review of the state of the art revealed the need for methods that allow the use of internal data generated from the use of intelligent products and their digital platforms, which become the means of interaction with the product and the services it offers. To address this need, we developed AdaptUI as a framework that leverages context-aware recommendation systems for the generation of adaptive user interfaces to enhance service delivery for users. To date, the creation of frameworks and methodologies for the development of AUI has been limited. Moreover, they do not integrate technological implementation with the evaluation of user experience and usability. This is due to the challenging nature of integrating these components that allow generalization into various scenarios. In this body of work, we have tried to narrow this gap by providing a framework that considers not only design aspects, but also technological implementation, and addresses aspects from data acquisition to adaptation and evaluation. The thesis objective, which aimed to demonstrate the usability and impact on the user experience in two scenarios— a commercial context for end users and an industrial environment—has been achieved.

Bibliography

Abowd Gregory D, Dey Anind K, Brown Peter J, Davies Nigel, Smith Mark, Steggles Pete. Towards a better understanding of context and context-awareness // International symposium on handheld and ubiquitous computing. 1999. 304–307.

Abrahão Silvia, Insfran Emilio, Sluÿters Arthur, Vanderdonckt Jean. Model-based intelligent user interface adaptation: challenges and future directions // Software and Systems Modeling. 2021. 20, 5. 1335–1349.

Adomavicius Gediminas, Tuzhilin Alexander. Context-aware recommender systems // Recommender systems handbook. 2010. 217–253.

Alcayaga Andres, Wiener Melanie, Hansen Erik G. Towards a framework of smart-circular systems: An integrative literature review // Journal of cleaner production. 2019. 221. 622–634. Publisher: Elsevier.

Alexopoulos Kosmas, Sipsas Konstantinos, Xanthakis Evangelos, Makris Sotiris, Mourtzis Dimitris. An industrial Internet of things based platform for context-aware information services in manufacturing // International Journal of Computer Integrated Manufacturing. 2018. 31, 11. 1111–1123. Publisher: Taylor & Francis.

Ali Mughees, Khan Saif Ur Rehman, Mashkoor Atif, Taskeen Anam. A conceptual framework for context-driven self-adaptive intelligent user interface based on Android // Cognition, Technology & Work. 2024. 26, 1. 83–106.

Alroobaea Roobaea, Mayhew Pam J. How many participants are really enough for usability studies? // 2014 Science and Information Conference. 2014. 48–56.

Araújo Harilton Da Silva, Rodrigues Joel JPC, Rabelo Ricardo De AL, Sousa Natanael De C, Sobral José VV, others . A proposal for IoT dynamic routes selection based on contextual information // Sensors. 2018. 18, 2. 353. Publisher: Multidisciplinary Digital Publishing Institute.

Arora Chetan, Sabetzadeh Mehrdad, Briand Lionel C, Zimmer Frank. Requirement boilerplates: Transition from manually-enforced to automatically-verifiable natural language patterns // 2014 IEEE 4th International Workshop on Requirements Patterns (RePa). 2014. 1–8.

Athanasopoulou Lydia, Papacharalampopoulos Alexios, Stavropoulos Panos. Context awareness system in the use phase of a smart mobility platform: A vision system for a light-weight approach // Procedia CIRP. 2020. 88. 560–564. Publisher: Elsevier.

Bailly Gilles, Lecolinet Eric, Nigay Laurence. Visual menu techniques // ACM Computing Surveys (CSUR). 2016. 49, 4. 1–41.

Belli Laura, Cirani Simone, Gorrieri Andrea, Picone Marco. A novel smart object-driven ui generation approach for mobile devices in the internet of things // Proceedings of the 1st international workshop on experiences with the design and implementation of smart objects. 2015. 1–6.

Berman Alexander, Thakare Ketan, Howell Joshua, Quek Francis, Kim Jeeeun. HowDIY: towards meta-design tools to support anyone to 3D print anywhere // 26th International Conference on Intelligent User Interfaces. 2021. 491–503.

Bettini Claudio, Brdiczka Oliver, Henricksen Karen, Indulska Jadwiga, Nicklas Daniela, Ranganathan Anand, Riboni Daniele. A survey of context modelling and reasoning techniques // Pervasive and mobile computing. 2010. 6, 2. 161–180. Publisher: Elsevier.

Bosch . Bosch Smart Home. 2022.

Brdnik Saša, Heričko Tjaša, Šumak Boštjan. Intelligent user interfaces and their evaluation: A systematic mapping study // Sensors. 2022. 22, 15. 5830.

Bu Lingguo, Chen Chun-Hsien, Ng Kam KH, Zheng Pai, Dong Guijun, Liu Heshan. A user-centric design approach for smart product-service systems using virtual reality: a case study // Journal of Cleaner Production. 2021. 280. 124413. Publisher: Elsevier.

Campos Pedro G, Bellogín Alejandro, Díez Fernando, Chavarriaga J Enrique. Simple time-biased KNN-based recommendations // Proceedings of the Workshop on Context-Aware Movie Recommendation. 2010. 20–23.

Carrera-Rivera Angela, Larrinaga Felix, Lasa Ganix. Context-awareness for the design of Smart-product service systems: Literature review // Computers in Industry. 2022. 142. 103730.

Carrera-Rivera Angela, Larrinaga Felix, Lasa Ganix. A context-aware framework for recommendation-based adaptive user interfaces in Smart Product-Service System. 2024. PREPRINT (Version 1) available at Research Square.

Carrera-Rivera Angela, Reguera-Bakhache Daniel, Larrinaga Felix, Lasa Ganix, Garitano Iñaki. Structured dataset of human-machine interactions enabling adaptive user interfaces // Scientific Data. 2023. 10, 1. 831.

Chang Danni, Gu Zhenyu, Li Fan, Jiang Rong. A user-centric smart product-service system development approach: A case study on medication management for the elderly // Advanced Engineering Informatics. 2019. 42. 100979.

Chiu Ming-Chuan, Huang Jih-Hung, Gupta Saraj, Akman Gulsen. Developing a personalized recommendation system in a smart product service system based on unsupervised learning model // Computers in Industry. 2021. 128. 103421.

Chou Jyh-Rong. A TRIZ-based product-service design approach for developing innovative products // Computers & Industrial Engineering. 2021. 161. 107608. Publisher: Elsevier.

Compton M, Barnaghi P, Bermudez L, Castro RG, Corcho O, Cox S, Graybeal John, Hauswirth M, Henson C, Herzog A, others . The ssn ontology of the semantic sensor networks incubator group // Journal of Web Semantics: Science, Services and Agents on the World Wide Web, ISSN. 2011. 1570–8268.

Cong Jing-chen, Chen Chun-Hsien, Zheng Pai, Li Xinyu, Wang Zuoxu. A holistic relook at engineering design methodologies for smart product-service systems development // Journal of Cleaner Production. XI 2020a. 272. 122737. Publisher: Elsevier BV.

Cong Jingchen, Chen Chun-Hsien, Zheng Pai. Design entropy theory: A new design methodology for smart PSS development // Advanced Engineering Informatics. 2020b. 45. 101124.

Cong Jingchen, Zheng Pai, Bian Yuan, Chen Chun-Hsien, Li Jianmin, Li Xinyu. A machine learning-based iterative design approach to automate user satisfaction degree prediction in smart product-service system // Computers and Industrial Engineering. 2022a. 165. Cited by: 20.

Cong Jingchen, Zheng Pai, Bian Yuan, Chen Chun-Hsien, Li Jianmin, Li Xinyu. A machine learning-based iterative design approach to automate user satisfaction degree prediction in smart product-service system // Computers & Industrial Engineering. 2022b. 165. 107939.

Costa Nina, Patrício Lia, Morelli Nicola, Magee Christopher L. Bringing service design to manufacturing companies: integrating PSS and service design approaches // Design Studies. 2018. 55. 112–145. Publisher: Elsevier.

De Nicola Antonio, Missikoff Michele, Navigli Roberto. A proposal for a unified process for ontology building: UPON // Database and Expert Systems Applications: 16th International Conference, DEXA 2005, Copenhagen, Denmark, August 22-26, 2005. Proceedings 16. 2005. 655–664.

Dinh Loan Thi Ngoc, Karmakar Gour, Kamruzzaman Joarder. A survey on context awareness in big data analytics for business applications // Knowledge and Information Systems. 2020. 62, 9. 3387–3415.

Dong Yuanfa, Liu Wenrong. Research on UX evaluation method of design concept under multi-modal experience scenario in the earlier design stages // International Journal on Interactive Design and Manufacturing (IJIDeM). 2018. 12, 2. 505–515. Publisher: Springer.

Dong Yuanfa, Zhu Rongzhen, Tian Qihua, Liu Wenrong, Peng Wei. A Scenario Interaction-centered Conceptual Information Model for UX Design of User-oriented Product-service System // Procedia CIRP. 2019. 83. 335–338. Publisher: Elsevier.

Dou Jinhua, Qin Jingyan. Research on User Mental Model Acquisition Based on Multidimensional Data Collaborative Analysis in Product Service System Innovation Process // International Conference on Engineering Psychology and Cognitive Ergonomics. 2017. 35–44.

Esheiba Laila, Elgammal Amal, Helal Iman MA, El-Sharkawi Mohamed E. A hybrid knowledge-based recommender for product-service systems mass customization // Information. 2021. 12, 8. 296.

Fernandez-Garcia Antonio Jesus, Iribarne Luis, Corral Antonio, Wang James Z. Evolving mashup interfaces using a distributed machine learning and model transformation methodology // On the Move to Meaningful Internet Systems: OTM 2015 Workshops: Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, EI2N, FBM, INBAST, ISDE, META4eS, and MSC 2015, Rhodes, Greece, October 26-30, 2015. Proceedings. 2015. 401–410.

Fitbit . Fitbit Premium services. 2023.

Frank Alejandro G, Mendes Glauco HS, Ayala Néstor F, Ghezzi Antonio. Servitization and Industry 4.0 convergence in the digital transformation of product firms: A business model innovation perspective // Technological Forecasting and Social Change. 2019. 141. 341–351.

Gan Runze, Liang Jiaming, Ahmad Bashar I, Godsill Simon. Modeling intent and destination prediction within a Bayesian framework: Predictive touch as a usecase // Data-Centric Engineering. 2020. 1. e12.

Gil Yolanda, Garijo Daniel, Khider Deborah, Knoblock Craig A, Ratnakar Varun, Osorio Maximiliano, Vargas Hernán, Pham Minh, Pujara Jay, Shbita Basel, others . Artificial intelligence for modeling complex systems: taming the complexity of expert models to improve decision making // ACM Transactions on Interactive Intelligent Systems. 2021. 11, 2. 1–49.

Gobert Camille, Todi Kashyap, Bailly Gilles, Oulasvirta Antti. SAM: a modular framework for self-adapting web menus // Proceedings of the 24th International Conference on Intelligent User Interfaces. 2019. 481–484.

Goel Deepti, Pahal Nisha, Jain Parul, Chaudhury Santanu. An ontology-driven context aware framework for smart traffic monitoring // 2017 IEEE Region 10 Symposium (TENSYMP). 2017. 1–5.

Gonçalves Taisa Guidini, Rocha Ana Regina Cavalcanti da. Development process for intelligent user interfaces: an initial approach // Proceedings of the XVIII Brazilian Symposium on Software Quality. 2019. 210–215.

Gong Chao. Human-machine interface: Design principles of visual information in human-machine interface design // 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics. 2. 2009. 262–265.

Gross KC, Baclawski K, Chan ES, Gawlick D, Ghoneimy A, Liu ZH. A supervisory control loop with prognostics for human-in-the-loop decision support and control applications // 2017 IEEE conference on cognitive and computational aspects of situation management (CogSIMA). 2017. 1–7.

Gruber Tom. What is an Ontology. 1993.

Guarino Nicola. Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy. 46. 1998.

Guarino Nicola, Oberle Daniel, Staab Steffen. What is an ontology? // Handbook on ontologies. 2009. 1–17.

Hajimohammadi Azadeh, Cavalcante Juliana, Gzara Lilia. Ontology for the PSS lifecycle management // Procedia CIRP. 2017. 64. 151–156. Publisher: Elsevier.

Hallstedt Sophie I, Isaksson Ola, Rönnbäck Anna Öhrwall. The Need for New Product Development Capabilities from Digitalization, Sustainability, and Servitization Trends // Sustainability. 2020. 12, 23. 10222.

Hassenzahl Marc, Tractinsky Noam. User experience-a research agenda // Behaviour & information technology. 2006. 25, 2. 91–97. Publisher: Taylor & Francis.

He Jiaxun, Zhang Shuang. How digitalized interactive platforms create new value for customers by integrating B2B and B2C models? An empirical study in China // Journal of Business Research. 2022. 142. 694–706.

Herlocker Jonathan L, Konstan Joseph A, Terveen Loren G, Riedl John T. Evaluating collaborative filtering recommender systems // ACM Transactions on Information Systems (TOIS). 2004. 22, 1. 5–53.

Horrocks Ian. OWL: A Description Logic Based Ontology Language // Principles and Practice of Constraint Programming - CP 2005. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. 5–8.

Hribernik Karl, Franke Marco, Klein Patrick, Thoben Klaus-Dieter, Coscia Eva. Towards a platform for integrating product usage information into innovative product-service design // 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC). VI 2017. 1407–1413.

Hu Peizhao, Indulska Jadwiga, Robinson Ricky. An autonomic context management system for pervasive computing // 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom). 2008. 213–223.

Hussain Jamil, Ul Hassan Anees, Muhammad Bilal Hafiz Syed, Ali Rahman, Afzal Muhammad, Hussain Shujaat, Bang Jaehun, Banos Oresti, Lee Sungyoung. Model-based adaptive user interface based on context and user experience evaluation // Journal on Multimodal User Interfaces. 2018. 12. 1–16.

Hynes Gearoid, Reynolds Vinny, Hauswirth Manfred. A context lifecycle for web-based context management services // European Conference on Smart Sensing and Context. 2009. 51–65.

Iqbal Muhammad Waseem, Ch Nadeem Ahmad, Shahzad Syed Khuram, Naqvi Muhammad Raza, Khan Babar Ayub, Ali Zulfiqar. User context ontology for adaptive mobile-phone interfaces // IEEE Access. 2021. 9. 96751–96762.

Iso DIS. 9241–210: 2010: ergonomics of human-system interaction—part 210: human-centred design for interactive systems (formerly known as 13407) // Switzerland: International Standards Organization. 2010.

Jakob Nielsen. Why you only need to test with 5 users // Nielsen Norman Group, Nielsen. 2000.

Jannach Dietmar, Zanker Markus, Felfernig Alexander, Friedrich Gerhard. Recommender systems: an introduction. 2010.

Janowicz Krzysztof, Haller Armin, Cox Simon JD, Le Phuoc Danh, Lefrançois Maxime. SOSA: A lightweight ontology for sensors, observations, samples, and actuators // Journal of Web Semantics. 2019. 56. 1–10.

Jia Guo, Zhang Guiyi, Yuan Xin, Gu Xiaosong, Liu Heshan, Fan Zhijun, Bu Lingguo. A synthetical development approach for rehabilitation assistive smart product–service systems: A case study // Advanced Engineering Informatics. 2021. 48. 101310. Publisher: Elsevier.

Johnston Vivien, Black Michaela, Wallace J. G. A Holistic UX Methodological Framework for Measuring the Aspects of How Dynamic, Adaptive and Intelligent a Software Solution is and Make Recommendations for Improvement // Collaborative European Research Conference (CERC 2020). VIII 2020.

Johnston Vivien, Black Michaela, Wallace Jonathan, Mulvenna Maurice, Bond Raymond. A framework for the development of a dynamic adaptive intelligent user interface to enhance the user experience // Proceedings of the 31st European Conference on Cognitive Ergonomics. 2019. 32–35.

Kaasinen Eija, Roto Virpi, Hakulinen Jaakko, Heimonen Tomi, Jokinen Jussi PP, Karvonen Hannu, Keskinen Tuuli, Koskinen Hanna, Lu Yichen, Saariluoma Pertti, others . Defining user experience goals to guide the design of industrial systems // Behaviour & Information Technology. 2015. 34, 10. 976–991. Publisher: Taylor & Francis.

Kaminskas Marius, Bridge Derek. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems // ACM Transactions on Interactive Intelligent Systems (TiiS). 2016. 7, 1. 1–42.

Kayes ASM, Rahayu Wenny, Dillon Tharam, Chang Elizabeth, Han Jun. Context-aware access control with imprecise context characterization through a combined fuzzy logic and ontology-based approach // OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". 2017. 132–153.

Keele Staffs, others . Guidelines for performing systematic literature reviews in software engineering. 2007.

Khan Inayat, Khusro Shah. Towards the design of context-aware adaptive user interfaces to minimize drivers' distractions // Mobile Information Systems. 2020. 2020.

Kim Yong Se. A FRAMEWORK FOR SMART EXPERIENCE DESIGN BASED ON ZERO-PARTY CUSTOMER EXPERIENCE DATA // Proceedings of the Design Society. 2023. 3. 161–170.

Kim Yong Se, Hong YeonKoo. A systematic method to design product-service systems using personalisation services based on experience evaluations // International Journal of Product Development. 2019. 23, 4. 353–386. Publisher: Inderscience Publishers (IEL).

Kiritsis Dimitris. Closed-loop PLM for intelligent products in the era of the Internet of things // Computer-Aided Design. 2011. 43, 5. 479–501. Publisher: Elsevier.

Kordestani Hossain, Mojarad Roghayeh, Chibani Abdelghani, Osmani Aomar, Amirat Yacine, Barkaoui Kamel, Zahran Wagdy. Hapicare: A healthcare monitoring system with self-adaptive coaching using probabilistic reasoning // 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). 2019. 1–8.

Kuhlenkötter Bernd, Wilkens Uta, Bender Beate, Abramovici Michael, Süße Thomas, Göbel Jens, Herzog Michael, Hypki Alfred, Lenkenhoff Kay. New perspectives for generating smart PSS solutions–life cycle, methodologies and transformation // Procedia CIRP. 2017. 64. 217–222. Publisher: Elsevier.

Kulkarni Saurabh, Rodd Sunil F. Context Aware Recommendation Systems: A review of the state of the art techniques // Computer Science Review. 2020. 37. 100255.

Kumar Naveen, Prajapati Shyambihari. Challenges for Interface Designers in Designing Sensor Dashboards in the Context of Industry 4.0 // International Journal of Industrial and Manufacturing Engineering. 2019. 13, 8. 539–542.

Landowska Agnieszka, Szwoch Mariusz, Szwoch Wioleta. Methodology of affective intervention design for intelligent systems // Interacting with Computers. 2016. 28, 6. 737–759.

Le Dinh Thang, Pham Thi Thanh Thoa, Pham-Nguyen Cuong, Nam Le Nguyen Hoai. A knowledge-based model for context-aware smart service systems // Journal of Information and Telecommunication. 2021. 1–22. Publisher: Taylor & Francis.

Le Dinh Thang, Thi Thanh Thoa Pham, Pham-Nguyen Cuong, Nam Le Nguyen Hoai. Towards a context-aware knowledge model for smart service systems // International Conference on Computational Collective Intelligence. 2020. 767–778.

Lenka Sambit, Parida Vinit, Wincent Joakim. Digitalization capabilities as enablers of value co-creation in servitizing firms // Psychology & marketing. 2017. 34, 1. 92–100. Publisher: Wiley Online Library.

Lenze . Lenze Smart Apps. 2022.

Li Gai, Wang Liyang, Ou Weihua. Robust personalized ranking from implicit feedback // International Journal of Pattern Recognition and Artificial Intelligence. 2016. 30, 01. 1659001.

Li Xinyu, Chen Chun-Hsien, Zheng Pai, Wang Zuoxu, Jiang Zuhua, Jiang Zhixing. A knowledge graph-aided concept–knowledge approach for evolutionary smart product–service system development // Journal of Mechanical Design. 2020a. 142, 10. 101403. Publisher: American Society of Mechanical Engineers.

Li Xinyu, Wang Zuoxu, Chen Chun-Hsien, Zheng Pai. A data-driven reversible framework for achieving Sustainable Smart product-service systems // Journal of Cleaner Production. I 2021. 279. 123618.

Li Xinyu, Wang Zuoxu, Zheng Pai, Peng Tao, Chen Chun-Hsien. A Closed-Loop Context-Aware Framework for Sustainable Smart PSS Development // 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE). VIII 2020b.

Lim Myung Joon, Kim Yong Se, others . Service design for people with disabilities using context-based activity modelling and international classification of functioning, disability and health // DS 87-3 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 3: Product, Services and Systems Design, Vancouver, Canada, 21-25.08. 2017. 2017. 231–240.

Lin Kuo-Yi, Chien Chen-Fu, Kerh Rhoann. UNISON framework of data-driven innovation for extracting user experience of product design of wearable devices // Computers & Industrial Engineering. 2016. 99. 487–502.

Lindgaard Gitte, Chattratichart Jarinee. Usability testing: what have we overlooked? // Proceedings of the SIGCHI conference on Human factors in computing systems. 2007. 1415–1424.

Liu Nathan N, Xiang Evan W, Zhao Min, Yang Qiang. Unifying explicit and implicit feedback for collaborative filtering // Proceedings of the 19th ACM international conference on Information and knowledge management. 2010. 1445–1448.

Liu Wei, Li Xue, Huang Daoli. A survey on context awareness // 2011 International Conference on Computer Science and Service System (CSSS). 2011. 144–147.

Liu Zhiwen, Ming Xinguo, Song Wenyan, Qiu Siqi, Qu Yuanju. A perspective on value co-creation-oriented framework for smart product-service system // Procedia Cirp. 2018. 73. 155–160. Publisher: Elsevier.

Mabuyo Michelle, Sullivan Megan. Interfaces. 2022.

Maleki Elaheh, Belkadi Farouk, Boli Nikoletta, Van Der Zwaag Berend Jan, Alexopoulos Kosmas, Koukas Spyridon, Marin-Perianu Mihai, Bernard Alain, Mourtzis Dimitris. Ontology-based framework enabling smart product-service systems: application of sensing systems for machine health monitoring // IEEE internet of things journal. 2018a. 5, 6. 4496–4505.

Maleki Elaheh, Belkadi Farouk, Boli Nikoletta, Van Der Zwaag Berend Jan, Alexopoulos Kosmas, Koukas Spyridon, Marin-Perianu Mihai, Bernard Alain, Mourtzis Dimitris. Ontology-based framework enabling smart product-service systems: application of sensing systems for machine health monitoring // IEEE internet of things journal. 2018b. 5, 6. 4496–4505. Publisher: IEEE.

Marczyk Geoffrey R, DeMatteo David, Festinger David. Essentials of research design and methodology. 2. 2010.

Martins Aline Freitas, Almeida Falbo Ricardo de. Models for Representing Task Ontologies. // WONTO. 2008.

McNee Sean M, Riedl John, Konstan Joseph A. Being accurate is not enough: how accuracy metrics have hurt recommender systems // CHI'06 extended abstracts on Human factors in computing systems. 2006. 1097–1101.

Mezhoudi Nesrine, Khaddam Iyad, Vanderdonckt Jean. Toward usable intelligent user interface // Human-Computer Interaction: Interaction Technologies: 17th International Conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part II 17. 2015. 459–471.

Michelson Brenda M. Event-driven architecture overview // Patricia Seybold Group. 2006. 2, 12. 10–1571.

Miraz Mahdi H, Ali Maaruf, Excell Peter S. Adaptive user interfaces and universal usability through plasticity of user interface design // Computer Science Review. 2021. 40. 100363.

Mitchell Jeffrey, Shneiderman Ben. Dynamic versus static menus: an exploratory comparison // ACM SigCHI Bulletin. 1989. 20, 4. 33–37.

Mizoguchi R, Vanwelkenhuysen J, Ikeda M. Task ontology: Ontology for building conceptual problem solving models/Towards Very Large Knowledge Bases. 1995.

Mourtzis Dimitris, Fotia Sophia, Vlachou Ekaterini, Koutoupes Angelos. A Lean PSS design and evaluation framework supported by KPI monitoring and context sensitivity tools // The International Journal of Advanced Manufacturing Technology. 2018. 94, 5-8. 1623–1637. Publisher: Springer Nature BV.

Nalpon Nicolas, Allignol Cyril, Picard Célia. Towards a User Interface Description Language Based on Bigraphs // International Colloquium on Theoretical Aspects of Computing. 2022. 360–368.

Ngoc Hien Nguyen, Lasa Ganix, Iriarte Ion. Human-centred design in industry 4.0: case study review and opportunities for future research // Journal of Intelligent Manufacturing. 2021. 1–42. Publisher: Springer.

Nguyen Loc. A new aware-context collaborative filtering approach by applying multivariate logistic regression model into general user pattern // Journal of Data Analysis and Information Processing. 2016. 4, 03. 124.

Nurmi Petteri, Floréen Patrik. Reasoning in context-aware systems // Helsinki Institute for Information Technology, Position paper. 2004. Publisher: Citeseer.

Ochoa William, Larrinaga Felix, Pérez Alain. Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches // Future Generation Computer Systems. 2023.

Oestreich Hendrik, Heinz-Jakobs Mario, Sehr Philip, Wrede Sebastian. Human-centered adaptive assistance systems for the shop floor // Human-Technology Interaction: Shaping the Future of Industrial User Interfaces. 2022. 83–125.

Ooijen Iris van, Vrabec Helena U. Does the GDPR enhance consumers' control over personal data? An analysis from a behavioural perspective // Journal of consumer policy. 2019. 42, 1. 91–107.

Orghidan Radu, Gordan Mihaela, Danciu Marius, Vlaicu Aurel. A prototype for the creation and interactive visualization of 3D human face models // Advanced Engineering Forum. 8. 2013. 45–54.

Ozok A Ant, Fan Quyin, Norcio Anthony F. Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: results from a college student population // Behaviour & Information Technology. 2010. 29, 1. 57–83.

Pan Zhiwen, Hariri Salim, Pacheco Jesus. Context aware intrusion detection for building automation systems // Computers & Security. 2019. 85. 181–201. Publisher: Elsevier.

Pandya Bhavesh, Pourabdollah Amir, Lotfi Ahmad. Fuzzy-as-a-Service for Real-Time Human Activity Recognition Using IEEE 1855-2016 Standard // 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). 2020. 1–8.

Papadakis Harris, Papagrigoriou Antonis, Panagiotakis Costas, Kosmas Eleftherios, Fragopoulou Paraskevi. Collaborative filtering recommender systems taxonomy // Knowledge and Information Systems. 2022. 64, 1. 35–74.

Peck Evan M, Easse Eleanor, Marshall Nick, Stratton William, Perrone L Felipe. Flyloop: a micro framework for rapid development of physiological computing systems // Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems. 2015. 152–157.

Peffers Ken, Tuunanen Tuure, Rothenberger Marcus A, Chatterjee Samir. A design science research methodology for information systems research // Journal of management information systems. 2007. 24, 3. 45–77.

Perera Charith, Zaslavsky Arkady, Christen Peter, Georgakopoulos Dimitrios. Context aware computing for the internet of things: A survey // IEEE communications surveys & tutorials. 2013. 16, 1. 414–454. Publisher: Ieee.

Petersen Kai, Vakkalanka Sairam, Kuzniarz Ludwik. Guidelines for conducting systematic mapping studies in software engineering: An update // Information and software technology. 2015. 64. 1–18.

Petticrew Mark, Roberts Helen. Systematic reviews in the social sciences: A practical guide. 2008.

Pezzotta Giuditta, Cavalieri Sergio, Romero David. Collaborative product-service systems engineering: Towards an active role of customers and stakeholders in value co-creation // 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC). 2017. 1247–1254.

Pingos Michalis, Elgammal Amal, Kumara Indika, Christodoulou Panayiotis, Papazoglou Mike P, Andreou Andreas S. Smart Shop-floor Monitoring via Manufacturing Blueprints and Complex-event Processing // International Conference on Enterprise Information Systems. 2019.

Pirola Fabiana, Boucher Xavier, Wiesner Stefan, Pezzotta Giuditta. Digital technologies in product-service systems: a literature review and a research agenda // Computers in Industry. 2020. 123. 103301.

Poveda-Villalón María, Gómez-Pérez Asunción, Suárez-Figueroa Mari Carmen. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation // International Journal on Semantic Web and Information Systems (IJSWIS). 2014. 10, 2. 7–34.

Pu Pearl, Chen Li, Hu Rong. A user-centric evaluation framework for recommender systems // Proceedings of the fifth ACM conference on Recommender systems. 2011. 157–164.

Pu Pearl, Chen Li, Hu Rong. Evaluating recommender systems from the user's perspective: survey of the state of the art // User Modeling and User-Adapted Interaction. 2012. 22. 317–355.

Pulikottil Terrin, Estrada-Jimenez Luis A, Abadía José Joaquín Peralta, Carrera-Rivera Angela, Torayev Agajan, Rehman Hamood Ur, Mo Fan, Nikghadam-Hojjati Sanaz, Barata Jose. Big Data Life Cycle in Shop-floor–Trends and Challenges // IEEE Access. 2023.

Rapaccini Mario, Saccani Nicola, Kowalkowski Christian, Paiola Marco, Adrodegari Federico. Navigating disruptive crises through service-led growth: The impact of COVID-19 on Italian manufacturing firms // Industrial Marketing Management. 2020. 88. 225–237. Publisher: Elsevier.

Reguera-Bakhache Daniel, Garitano Iñaki, Uribeetxeberria Roberto, Cernuda Carlos, Zurutuza Urko. Data-driven industrial human-machine interface temporal adaptation for process optimization // 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). 1. 2020. 518–525.

Ren Mengyang, Dong Liang, Xia Ziqing, Cong Jingchen, Zheng Pai. A Proactive Interaction Design Method for Personalized User Context Prediction in Smart-Product Service System // Procedia CIRP. 2023. 119. 963–968. The 33rd CIRP Design Conference.

Ren Shan, Zhang Yingfeng, Sakao Tomohiko, Liu Yang, Cai Ruilong. An Advanced Operation Mode with Product-Service System Using Lifecycle Big Data and Deep Learning // International Journal of Precision Engineering and Manufacturing-Green Technology. 2021. 1–17. Publisher: Springer.

Ren Yuan, Parvizi Artemis, Mellish Chris, Pan Jeff Z, Van Deemter Kees, Stevens Robert. Towards competency question-driven ontology authoring // The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings 11. 2014. 752–767.

Ricci Francesco, Rokach Lior, Shapira Bracha. Recommender Systems Handbook // Recommender Systems Handbook. Boston, MA: Springer US, 2011. 1–35.

Ritter Marli, Winterbottom Cara. UX for the Web: Build websites for user experience and usability. 2017.

Rodrigues João MF, Pereira João AR, Sardo João DP, Freitas Marco AG de, Cardoso Pedro JS, Gomes Miguel, Bica Paulo. Adaptive card design UI implementation for an augmented reality museum application // 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. 2017. 433–443.

Seo Dong Woo, Kim Hyun, Kim Jae Sung, Lee Jae Yeol. Hybrid reality-based user experience and evaluation of a context-aware smart home // Computers in Industry. 2016. 76. 11–23. Publisher: Elsevier.

Simple Complex . Parsif.al. 2021.

Smirnov Alexander V, Kashevnik Alexey M, Ponomarev Andrew. Context-based infomobility system for cultural heritage recommendation: Tourist Assistant—TAIS // Personal and Ubiquitous Computing. 2017. 21. 297–311.

Spear Andrew D, Ceusters Werner, Smith Barry. Functions in basic formal ontology // Applied Ontology. 2016. 11, 2. 103–128. Publisher: IOS Press.

Stefanidi Zinovia, Margetis George, Ntoa Stavroula, Papagiannakis George. Real-Time Adaptation of Context-Aware Intelligent User Interfaces, for Enhanced Situational Awareness // IEEE Access. 2022. 10. 23367–23393.

Stephanidis Constantine. Adaptive techniques for universal access // User modeling and user-adapted interaction. 2001. 11. 159–179.

Stephanidis Constantine, Paramythis Alex, Sfyrakis Michael, Stergiou A, Maou Napoleon, Leventis A, Paparoulis George, Karagiannidis Charalampos. Adaptable and adaptive user interfaces for disabled users in the AVANTI project // 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. 1998. 153–166.

Stumpf Simone. Horses for courses: Making the case for persuasive engagement in smart systems // Joint Proceedings of the ACM IUI 2019 Workshops. 2327. 2019.

Tahir Rabail. Analyzing the intelligence in user interfaces // 2015 SAI Intelligent Systems Conference (IntelliSys). 2015. 674–680.

Tan Hua-Zhe, Zhao Wei, Shen Hai-Hua. Adaptive user interface optimization for multi-screen based on machine learning // 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)). 2018. 743–748.

Tarus John K, Niu Zhendong, Yousif Abdallah. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining // Future Generation Computer Systems. 2017. 72. 37–48.

ThoughtWorks . ThoughtWorks Technology Radar, Vol. 27. 2022. [Online; accessed 19-June-2023].

Tsandilas Theophanis, Schraefel MC. An empirical assessment of adaptation techniques // CHI'05 Extended Abstracts on Human Factors in Computing Systems. 2005. 2009–2012.

Tukker A. Eight types of product–service system: eight ways to sustainability? Experiences from SusProNet // Business Strategy and The Environment. 2004. 13. 246–260.

Valencia Ana, Mugge Ruth, Schoormans Jan, Schifferstein Hendrik. The design of smart product-service systems (PSSs): An exploration of design characteristics // International Journal of Design. 2015. 9, 1.

Valencia Cardona AM, Mugge Ruth, Schoormans Jan PL, Schifferstein Hendrik NJ. Challenges in the design of smart product-service systems (PSSs): Experiences from practitioners // Proceedings of the 19th DMI: Academic Design Management Conference. Design Management in an Era of Disruption, London, UK, September 2-4, 2014. 2014.

Van Eck Nees Jan, Waltman Ludo. Software survey: VOSviewer, a computer program for bibliometric mapping // scientometrics. 2010. 84, 2. 523–538. Publisher: Springer.

Vandermerwe Sandra, Rada Juan. Servitization of business: adding value by adding services // European management journal. 1988. 6, 4. 314–324. Publisher: Elsevier.

Völkel Sarah Theres, Schneegass Christina, Eiband Malin, Buschek Daniel. What is" intelligent" in intelligent user interfaces? a meta-analysis of 25 years of IUI // Proceedings of the 25th international conference on intelligent user interfaces. 2020. 477–487.

Wang Zuoxu, Chen Chun-Hsien, Li Xinyu, Zheng Pai, Khoo Li Pheng. A context-aware concept evaluation approach based on user experiences for smart product-service systems design iteration // Advanced Engineering Informatics. 2021a. 50. 101394. Publisher: Elsevier.

Wang Zuoxu, Chen Chun-Hsien, Zheng Pai, Li Xinyu, Khoo Li Pheng. A novel data-driven graph-based requirement elicitation framework in the smart product-service system context // Advanced engineering informatics. 2019a. 42. 100983. Publisher: Elsevier.

Wang Zuoxu, Chen Chun-Hsien, Zheng Pai, Li Xinyu, Khoo Li Pheng. A graph-based context-aware requirement elicitation approach in smart product-service systems // International Journal of Production Research. 2021b. 59, 2. 635–651.

Wang Zuoxu, Chen Chun-Hsien, Zheng Pai, Li Xinyu, Song Wenyan. A hypergraph-based approach for context-aware smart product-service system configuration // Computers & Industrial Engineering. 2021c. 107816.

Wang Zuoxu, Li Xinyu, Chen Chun-Hsien, Khoo Li-Pheng. Evaluating Smart PSS Solutions with Context-Awareness in Usage Phase // Transdisciplinary Engineering for Complex Socio-technical Systems – Real-life Applications. 12. IX 2020. 333–342. (Advances in Transdisciplinary Engineering). ISSN: 2352751X, 23527528.

Wang Zuoxu, Zheng Pai, Chen Chun-Hsien, Khoo Li Pheng. A Survey of Requirements Management in Smart Product-Service Systems // Transdisciplinary Engineering for Complex Socio-technical Systems. 2019b. 613–622.

Wenjuan Hu, Premalatha R, Aiswarya RS. Physical education system and training framework based on human–computer interaction for augmentative and alternative communication // International Journal of Speech Technology. 2022. 1–11.

Wetter-Edman Katarina, Sangiorgi Daniela, Edvardsson Bo, Holmlid Stefan, Grönroos Christian, Mattelmäki Tuuli. Design for value co-creation: Exploring synergies between design for service and service logic // Service Science. 2014. 6, 2. 106–121. Publisher: INFORMS.

Wohlin Claes, Runeson Per, Höst Martin, Ohlsson Magnus C., Regnell Björn, Wesslén Anders. Systematic Literature Reviews // Experimentation in Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. 45–54.

Wuest Thorsten, Wellsandt Stefan. Design and Development of Product Service Systems (PSS)-Impact on Product Lifecycle Perspective // Procedia Technology. 2016. 26. 152–161. Publisher: Elsevier.

Wutzler Markus, Weissbach Martin, Springer Thomas. Role-based models for building adaptable collaborative smart service systems // 2017 IEEE International Conference on Smart Computing (SMARTCOMP). 2017. 1–6.

Yu Ying, Sung Tung-Jung. A value-based view of the smart PSS adoption: a study of smart kitchen appliances // Service Business. 2023. 1–29.

Yuan Wenyu, Chang Danni, Han Ting. A context-aware smart product-service system development approach and application case // Computers & Industrial Engineering. 2023. 183. 109468.

Zambetti Michela Giuseppina, Federico Adrodegari, Pezzotta Giuditta, Pinto Roberto, Nicola Saccani. Understanding Data-Driven Product Service System characteristics: a conceptual framework for manufacturing applications // XXV Summer School "Francesco Turco"–Industrial Systems Engineering. 2020. N–A.

Zhang Xianyu, Chen LuCheng, Sheng GuoJun, Lu XiaoPing, Ming Xinguo. An innovation service system and personalized recommendation for customer-product interaction life cycle in smart product service system // Journal of Cleaner Production. 2023. 398. 136470.

Zheng Chen, Wang Zhanxi, Qin Xiansheng, Eynard Benoît, Hehenberger Peter, Li Jing, Bai Jing, Zhang Yicha. Integrated design for product–service systems: a focus on multi-disciplinary interface // International Journal of Production Research. 2021. 59, 19. 5884–5902.

Zheng Pai, Liu Yang, Tao Fei, Wang Zuoxu, Chen Chun-Hsien. Smart product-service systems solution design via hybrid crowd sensing approach // IEEE Access. 2019a. 7. 128463–128473.

Zheng Pai, Wang Zuoxu, Chen Chun-Hsien, Khoo Li Pheng. A survey of smart product-service systems: Key aspects, challenges and future perspectives // Advanced engineering informatics. 2019b. 42. 100973. Publisher: Elsevier.

Zheng Pai, Yu Shiqiang, Wang Yuanbin, Zhong Ray Y, Xu Xun. User-experience based product development for mass personalization: a case study // Procedia CIRP. 2017. 63. 2–7. Publisher: Elsevier.

Zhou Jianlong, Sun Jinjun, Chen Fang, Wang Yang, Taib Ronnie, Khawaji Ahmad, Li Zhidong. Measurable decision making with GSR and pupillary analysis for intelligent user interface // ACM Transactions on Computer-Human Interaction (ToCHI). 2015. 21, 6. 1–23.

Zhou Tongtong, Chen Zhihua, Cao Yong, Miao Rui, Ming Xinguo. An integrated framework of user experience-oriented smart service requirement analysis for smart product service system development // Advanced Engineering Informatics. 2022a. 51. 101458.

Zhou Tongtong, Chen Zhihua, Ming Xinguo. Multi-criteria evaluation of smart product-service design concept under hesitant fuzzy linguistic environment: A novel cloud envelopment analysis approach // Engineering Applications of Artificial Intelligence. 2022b. 115. Cited by: 5.

Zhou Tongtong, Ming Xinguo, Han Ting, Bao Yuguang, Liao Xiaoqiang, Tong Qingfei, Liu Shangwen, Guan Hao, Chen Zhihua. Smart experience-oriented customer requirement analysis for smart product service system: A novel hesitant fuzzy linguistic cloud DEMATEL method // Advanced Engineering Informatics. 2023. 56. Cited by: 4.

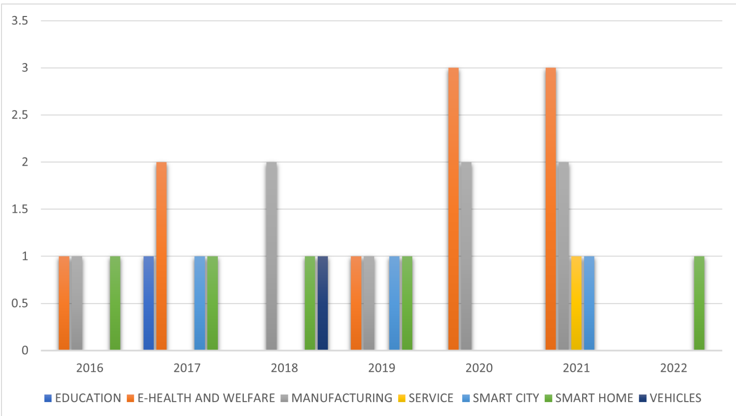
Zhou Xiang, Peng Xin, Xie Tao, Sun Jun, Ji Chao, Liu Dewei, Xiang Qilin, He Chuan. Latent error prediction and fault localization for microservice applications by learning from system trace logs // Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2019. 683–694.

Ziarani Reza Jafari, Ravanmehr Reza. Serendipity in recommender systems: a systematic literature review // Journal of Computer Science and Technology. 2021. 36. 375–396.

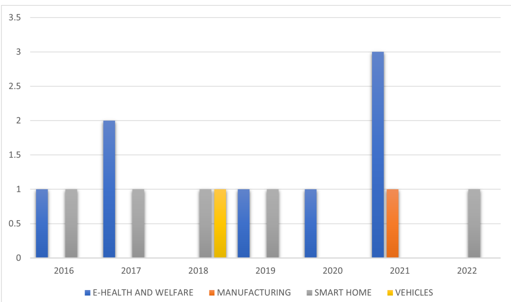
A Appendix

A.1 Supplementary yearly data

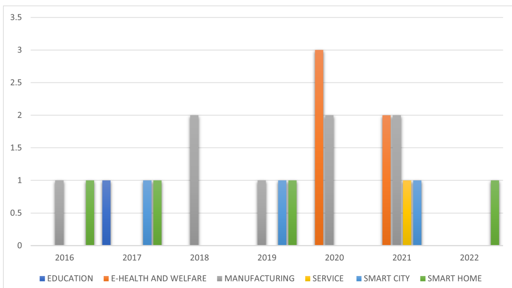
Figure A.1: Industrial Application Sector yearly trend



(a) Industrial Application Sector yearly trend



(b) Application Sector in UCD, yearly trend in S-PSS



(c) Application Sector in Context Awareness applications, yearly trend in S-PSS

A.2 Case Studies Empirical Articles

Reference	Solution Type	Industry	CA	UCD	Businness Type	Data Driven	PSS Type	Description
Lim et al. (2017)	FW	E-Health and welfare	N	Y	B2C	N	USE-ORIENTED	PSS DESIGN FOR THE KNEELING BUS, theoretical framework for Service design for people with disabilities
Pingos et al. (2019)	FW	Manufacturing	Y	N	B2B	Y	USE-ORIENTED	SHOP-FLOOR MONITORING, framework adopts a smart PSS lifecycle to offer a monitoring feedback loop that enables continuous product and Service improvements
Zambetti et al. (2020)	FW	Manufacturing	N	N	B2B	Y	PRODUCT-ORIENTED	Conceptual framework for identification of data needs in different dimensions
Wang et al. (2020, 2021a)	DA	Manufacturing	Y	N	B2C	Y	USE-ORIENTED	REMOTE 3D PRINTING Service, context aware approach to evaluate PSS bundles in diferent scenarios
Le Dinh et al. (2021)	DM	Service	Y	N	B2C	Y	PRODUCT-ORIENTED	SMART CHATBOT Knowledge model for context-aware smart Service systems based on the knowledge components, called the CAK (Context-Aware Knowledge) model
Chou (2021)	DA	E-Health and welfare	N	Y	B2C	N	USE-ORIENTED	SMART MASK, TRIZ-based product-Service design approach to assist designers/engineers in developing innovative products
Wutzler et al. (2017)	DM	Education	Y	N	B2C	N	USE-ORIENTED	INTERACTIVE CLASROOM, context-aware, role-based collaboration specification to model adaptable, collaborative smart Service systems
Dong et al. (2019)	DM	SMART HOME	Y	Y	B2C	N	PRODUCT-ORIENTED	SMART SWEEPING ROBOT, Scenario Interaction-centered Conceptual Information Model for UX
Zheng et al. (2021)	MT	Manufacturing	N	N	B2B	Y	PRODUCT-ORIENTED	ROBOTIC ENGRAVING SYSTEM, An integrated design method using an improved interface modelling approach that provides a structural form to represent the information transfer between product and Service components
Athanasopoulou et al. (2020)	DA	E-Health and welfare	Y	N	B2C	N	USE-ORIENTED	SMART MOBILITY PLATFORM, framework for the development of a context awareness and path planning system

Li et al. (2020b, 2021)	FW	Manufacturing	Y	N	B2C	Y	PRODUCT-ORIENTED	SMART 3D PRINTER, integrated CONTEXT-AWARE framework for sustainable Smart PSS development,
Mourtzis et al. (2018)	MT	Manufacturing	Y	N	B2B	Y	PRODUCT-ORIENTED	MOLD MAKING MACHINE, conceptual methodology for PSS evaluation and PSS Lean design assistance via context sensitivity for the selection of appropriate set of KPIs and lean rules to the PSS design.
Cong et al. (2020b)	MT	E-Health and welfare	Y	Y	B2C	Y	PRODUCT-ORIENTED	SMART TRAVEL ASSISTANCE for elderly, design entropy theory to determine the best design/redesign solutions
Zheng et al. (2019a)	DA	Smart city	Y	N	B2C	Y	USE-ORIENTED	SMART WATER DISPENSER, hybrid crowd sensing approach to enable design innovation in the Smart PSS context cost-effectively.
Hajimohammadi et al. (2017)	ON	SMART CITY	Y	N	B2C	N	PRODUCT-ORIENTED	SMART BIKE MAINTENANCE, Ontology PSS lifecycle
Wang et al. (2021b)	DA	Smart city	Y	N	B2C	S	USE-ORIENTED	SMART BIKE, Systematic context-aware requirement elicitation method **FOR PAPER) apply the extracted stakeholder requirements to assist stakeholders for decision makings (e.g. reconfigure the functions of the product-Service bundles
Chang et al. (2019)	DA	E-Health and welfare	N	Y	B2C	N	PRODUCT-ORIENTED	SMART PILLBOX for elderly, development approach for S-PSS from user-centric perspective
Li et al. (2020a)	DA	E-Health and welfare	Y	N	B2C	N	PRODUCT-ORIENTED	SMART NURSING BED, KG-CK approach to cost-effectively assist Smart PSS development on the foundation of knowledge graph techniques and concept–knowledge theory.
Jia et al. (2021)	DA	E-Health and welfare	Y	Y	B2C	Y	PRODUCT-ORIENTED	SMART REHABILITATION ASSISTIVE DEVICE, system development method for RAD that uses user-generated rehabilitation data, user experience data and user research data
Zhou et al. (2022a)	FW	Smart Home	Y	Y	B2C	Y	USE-ORIENTED	SMART HOME Service SYSTEM, methodological framework of user experience-oriented smart Service requirement (UXO-SSR) analysis for smart PSS development
Dou, Qin (2017)	MT	Smart Home	Y	Y	B2C	Y	PRODUCT-ORIENTED	SMART TV, method for acquiring user mental models based on multidimensional data synchronization analysis.

Bu et al. (2021)	DA	E-Health and welfare	Y	Y	B2C	Y		SMART VR ROW MACHINE, conceptual framework for a VR-assisted, user-centric, smart PSS
Mourtzis et al. (2018)	FW	Manufacturing	Y	N	B2B	N	PRODUCT-ORIENTED	MOLD MAKING MACHINE, conceptual methodology for PSS evaluation and PSS Lean design assistance via context sensitivity for the selection of appropriate set of KPIs and lean rules to the PSS design.
Maleki et al. (2018b)	FW	Manufacturing	Y	N	B2B	N	USE-ORIENTED	SMART HEALTH Manufacturing, ontology-based framework enabling smart product-Service systems: application of sensing systems for machine health monitoring
Zheng et al. (2017)	DA	E-Health and welfare	N	Y	B2C	Y	PRODUCT-ORIENTED	SMART MASK, three-model based (i.e. physical model, cyber model and user experience model) generic framework for conducting user experience based product development for mass personalization
Seo et al. (2016)	FW	Smart Home	Y	Y	B2C	Y	PRODUCT-ORIENTED	SMART HOME, hybrid reality-based user experience and evaluation of a context-aware smart home
Dong, Liu (2018)	DA	Vehicles	N	Y	B2C	N	PRODUCT-ORIENTED	SMART VEHICLE, multi-modal and scenario-based UX evaluation method of design concept in the earlier design stages
Liu et al. (2018)	FW	Smart Home	N	Y	B2B	Y	PRODUCT-ORIENTED	SMART REFRIGERATOR, value co-creation process for smart PSS, in which there are four stages of coexist, co-design, co-implement and co-evaluate
Lin et al. (2016)	FW	E-Health and welfare	N	Y	B2C	Y	PRODUCT-ORIENTED	SMART WEARABLE DEVICE, framework for data-driven innovation to capture the user experience and preference among the factors of product form designs to derive useful rules